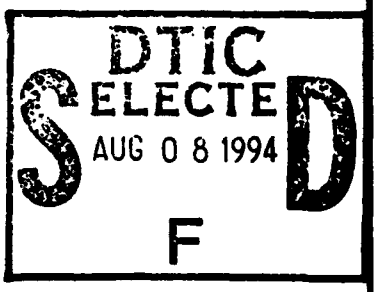
TEC-0046



Representation, Modeling and Recognition of Outdoor Scenes First Annual Report

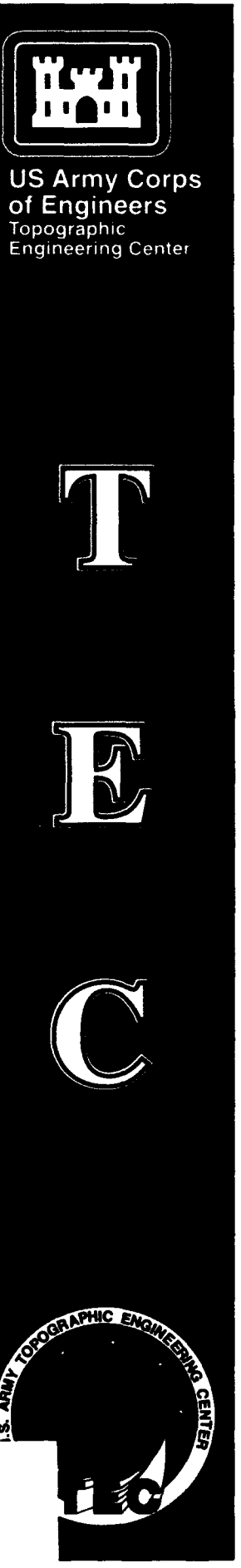
Martin A. Fischler
Robert C. Bolles

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493

November 1993

Prepared for:
Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

Monitored by:
U.S. Army Corps of Engineers
Topographic Engineering Center
7701 Telegraph Road
Alexandria, Virginia 22315-3864



Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1993	First Annual Report Mar. 1992 - Mar. 1993

4. TITLE AND SUBTITLE

Representation, Modeling and Recognition of Outdoor Scenec

5. FUNDING NUMBERS

DACA76-92-C-0008

6. AUTHOR(S)

Martin A. Fischler Robert C. Bolles

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Advanced Research Projects Agency
3701 North Fairfax Drive, Arlington, VA 22203-1714

U.S. Army Topographic Engineering Center
7701 Telegraph Road., Alexandria, VA 22315-3864

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

TEC-0046

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The primary goal of this project was to advance the state-of-the-art in scene interpretation for autonomous systems that operate in natural terrain. In particular, techniques are being developed for representing knowledge about complex cultural and natural environments so that a computer vision system can successfully plan, navigate, recognize and manipulate objects and answer questions or make decisions relevant to this knowledge. The initial results are centered on the development of representations and associated methods for rapidly modeling natural terrain (from image sequences) at a level of organization higher than that of the conventional dense array of depths. This work will provide the essential advance needed to turn raw geometric measurements into timely information usable by robotic navigation and planning systems. Work is also progressing on two additional problems: modeling compact 3-D objects from their projected 2-D contours, and the problem of recognizing important classes of natural and man-made objects -- especially roads, trees and rocks.

14. SUBJECT TERMS

Machine Vision, Automated Scene Analysis, Interactive Scene Analysis, Object Recognition, Terrain Modeling, Automated Cartography, Feature Extraction, Delineation, Partitioning, Geometric Modeling

15. NUMBER OF PAGES

71

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

PREFACE

This research is sponsored by the Advanced Research Projects Agency (ARPA) and monitored by the U.S. Army Topographic Engineering Center (TEC) under Contract DACA76-92-C-0008, titled "Representation, Modeling and Recognition of Outdoor Scenes, First Annual Report". The ARPA Program Manager is Dr. Oscar Firschein, and the TEC Contracting Officer's Representative is Ms. Lauretta Williams.

REPRESENTATION, MODELING AND RECOGNITION OF OUTDOOR SCENES

Martin A. Fischler and Robert C. Bolles
Principal Investigators

OBJECTIVE:

Our primary goal in this project is to advance the state of the art in scene interpretation for autonomous systems that operate in natural terrain. In particular, techniques are being developed for representing knowledge about complex cultural and natural environments so that a computer vision system can successfully plan, navigate, recognize, and manipulate objects and answer questions or make decisions relevant to this knowledge.

APPROACH:

This work integrates advances in four separate technologies to achieve the goal of providing a foundation for the design of highly competent machine vision systems capable of autonomous operation in the outdoor world.

First, stored knowledge (such as map data and object models) provides the basis for invoking context, function, and purpose, in addition to the use of visually observed geometric shape, to recognize scene objects.

Second, we are developing compact and expressive representations for modeling, and ultimately recognizing, objects encountered in the natural world. Computational efficiency, and thus real time performance, is critically dependent on using effective representations for both models and sensed data.

Third, global optimization techniques are being developed that require reasonable amounts of computation, but which are expected to produce results beyond those obtainable by local analysis methods.

Fourth, techniques are being developed that are able to simultaneously, or incrementally, exploit multiple views of a scene in compiling a complete scene model. For example, in our previous work we have been able to demonstrate that the integrated analysis of a motion sequence can be used to construct a geometric scene model that is superior to a sequence of independent stereo reconstructions.

PROGRESS:

This program builds on our previous ARPA research. Our initial results are centered on the development of representations and associated methods for rapidly modeling natural terrain (from image sequences) at a level of organization higher than that of the conventional dense array of depths. This work will provide the essential advance needed to turn raw geometric measurements into timely information usable by robotic navigation and planning systems. Work is also progressing on two additional problems: modeling compact 3-D objects from their projected 2-D contours, and the problem of recognizing important classes of natural and man-made objects -- especially roads, trees, and rocks.

SUMMARY OF RECENT ACCOMPLISHMENTS:

-Developed an approach for integration of information acquired from multiple views of a scene into a description of scene geometry. The approach uses a new class of geometric primitives which allows easy expression of known constraints and observed data, and also allows the use of practical optimization based solution techniques. This work will provide an effective way of allowing a robotic system to incrementally build a progressively more accurate and complete model of the environment in which it is operating. A paper describing this work, intended for journal publication, has been completed and is included in this report as Appendix A.; also see the detailed discussion of this topic provided in a following section.

-Made a significant new advance in the long-standing problem of duplicating human performance in recovering 3-D models of terrain and man-made objects from qualitative and imprecise line drawings (e.g., of terrain elevations as in an

approximate and uncalibrated contour map, or building edges as in a single approximate projection of the corresponding wire-frame). This work can greatly simplify communication problems between man and machine in such applications as robotic mission planning and in construction of databases for use in robotic navigation. A paper describing this work has been published in the International Journal of Computer Vision ("An optimization based approach to the interpretation of single line drawings as 3-D wire frames," IJCV 9(2):113-136, Nov 1992); a reprint is enclosed as Appendix B. On-going work has led to (new) additional results of both theoretical and practical importance; these new results will be described in a later report.

-The problem of automatically recognizing objects appearing in images of the outdoor world has proven to be extremely difficult, in part, because in addition to all the other difficulties of object recognition, we must now also contend with the lack of explicit shape models. While most of the current (successful) computer-based recognition approaches rely on explicit knowledge of shape, rocks, trees, and other natural objects cannot be successfully described in this way; even such generic man-made objects as roads, bridges, and buildings are more likely to satisfy functional constraints rather than being exemplars of some geometric blueprint. In order to replace explicit shape with a more general way of describing natural objects (and complex man-made structures), a large number of geometric primitives have been proposed that are also suitable for detection by automatic image analysis algorithms (e.g., edges, textures, fractals). The result of much of this past work is that, while often promising, the techniques are not sufficiently reliable to provide a basis for the knowledge-based analysis needed to complete the recognition task. What is required are a few techniques than can very reliably organize the pixel-level image data as a basis for higher level analysis. Finding the appropriate combination of low-level data-description, and associated extraction techniques, is thus a key problem in machine vision and of our primary concerns in this project. In addition to our work relevant to this topic discussed above, we have focused on extracting coherent line (as distinct from edge) features in single gray-level images. We note that a line sketch of some object or scene is often sufficient to depict the imaged information in a very compact way. Two techniques have emerged from this work that appear to meet the criterion of generality and robustness. The first is a generic way to find candidate line structure in an image; this work will be described in a later report. The second is a way to organize such data into perceptually coherent and semantically meaningful units. In Appendix C of this report we describe our progress in the design of a curve partitioning technique that is extremely robust in achieving the perceptual organization task; we also describe how this technique can be applied to the problem of road delineation in aerial images.

DETAILED DISCUSSION OF RECENT WORK ON GEOMETRIC RECONSTRUCTION FROM MULTIPLE VIEWS:

To reconstruct object surfaces, one can start with a number of measuring techniques, for example laser rangefinding, stereo or 3D scanners, all of which provide raw information about the location of points in space. These points, however, often form potentially noisy "clouds" of data instead of the surfaces one expects.

Deriving the surfaces from such data is a difficult task because:
-the 3D points may form a very irregular sampling of the space,
-they may have been produced by several sensors or derived from several viewpoints so that it becomes impossible to work only in the imaging plane of any one sensor,
-several surfaces can overlap; simple interpolation will not work,
-the sensors and algorithms make mistakes that must be properly dealt with.

In this research effort, we address the problem of determining the 3-D shape and material properties of surfaces by combining the information provided by active or passive ranging techniques with that present in multiple 2-D intensity images. As discussed in our previous reports, we are investigating two different approaches, the first based on local surfaces and the second on global ones.

At present, most of our efforts have been devoted to the global surface approach. It relies on hexagonal triangulations that can be deformed to recover both the geometry and physical properties of surfaces of interest.

Surface Geometry

ven camera models for the images being analyzed, the corresponding projections
the 3-D surface points appearing in the images can be computed and, assuming
e usual stereo assumption, must have comparable grey levels. Our algorithm
ptimizes the placement of surface vertices to minimize the overall difference
grey levels while preserving surface smoothness. The actual criterion we use
a linear combination of the sums of the variance of grey levels across images
d of the sums of the surface curvatures at the vertices. We use a
njugate-gradient descent algorithm embedded in a continuation method to
rform the optimization: we first optimize with a strong smoothness
nstraint; we then reduce the constraint progressively.

ecause our surfaces are 3-D objects, we can directly determine the presence of
dden surfaces and deal effectively with occlusions. In order to detect those
dden surfaces in an effective manner, we have implemented the algorithm to run
n an SGI machine and exploit the machines z-buffering capabilities.

far, in most of our experiments, we have used regular grids and uniform
oothness constraints. While this is appropriate for surfaces whose properties
main relatively constant, this is suboptimal for more complex surfaces that
n be more effectively handled using triangulated irregular networks. The
latively smooth parts of such surfaces should be represented by large patches
ile the rougher parts are better described by finer and less constrained
iangulations. We have made progress in implementing such irregular networks
allowing some of the regular facets to be subdivided as required by the
rface geometry.

) Physical Properties

any natural surfaces can be modeled by a Lambertian reflectance model whose
lbedo depends on the corresponding physical surface properties. Recovering
his albedo is therefore an important first step towards the goal of analyzing
hose physical properties and potentially segmenting regions of interest.
nlike traditional "shape from shading" approaches that work in image space and
ssume constant albedo, our technique allows us to assign different albedoes to
he facets of the derived triangulation. We can then optimize the values
ssigned to these albedoes and also find (or use the known) location of the
ight source to maximize the similarity between the shaded image derived from our
odels and the real images.

e are performing experiments with the above method for computing albedo given
rfaces originally derived using stereo. The objective function we optimize
nforces albedo smoothness while minimizing intensity difference between the
haded images and the real ones. To make this approach fully general, we will
ntroduce albedo discontinuities to account for abrupt changes in surface
aterial type. We will also attempt to determine those classes of natural
bjects and terrain types for which the Lambertian model is appropriate by
xamining the variance in intensity across images of the same scene acquired
rom different viewpoints.

ur ultimate goal in the above two tasks is to be able to optimize
imultaneously the vertex positions and the surface albedoes in order to compute
rface geometry and photometry. Our current focus in this task is to combine
he stereo objective function with the photometric one in order to achieve a
ore complete description of the scene.

) Implementation and Testing

n the past few months we have refined and tested our method for reconstructing
oth the shape and reflectance properties of physical surfaces from the
nformation present in multiple images. We have, so far, considered two classes
f information. The first class contains the information that can be extracted
rom a single image, such as texture gradients, shading, and occlusion edges.
e take advantage of the fact that multiple images enhance the utility of this
ype of information by allowing for consistency checks across the images as well
s the use of averaging to improve precision. The second class contains
nformation that require at least two images for its extraction, such as the
epth of corresponding points found in two input images through the use of
tereo triangulation.

ur surface reconstruction method uses an object-centered representation,
pecifically, a hexagonally-connected 3-D mesh of vertices with triangular

facets. Such a representation accommodates the two classes of information mentioned above, as well as multiple images (including motion sequences of a rigid object) and self-occlusions. We have chosen to model the surface material using the Lambertian reflectance model with variable albedo, though generalizations to specular surfaces are possible. Consequently, the natural choice for the monocular information source is shading, while intensity is the natural choice for the image feature used in multi-image correspondence. Not only are these the natural choices when we are able to assume a Lambertian reflectance model, they are complementary: intensity correlation is most accurate wherever the input images are highly textured, and shading is most accurate when the input images have smooth intensity variation. Since we wish to deal with surfaces with non-uniform albedo, we have developed a new approach to incorporating shading information that uses the variation in computed albedo from facet to facet as the indicator of a correct surface reconstruction.

We use an optimization approach to reconstruct the surface shape and its material properties from the input images. That is, we alter the shape and reflectance properties of the surface mesh so as to minimize an objective function, given an initial surface estimate provided by other means, such as a standard stereo algorithm. The objective function is a linear combination of an intensity correlation component, an albedo variation component, and a surface smoothness component. The first two components are a function of the intensities projected onto the triangular facets from the input images (taking occlusions into account), and are weighted according to the amount of texture in the intensities, for the reasons mentioned in the previous paragraph. The geometric smoothness component is slowly decreased during the optimization process to allow for an accurate estimate of the surface shape and reflectance.

We have implemented an algorithm employing these three terms and have performed extensive experiments using synthetic images as well as real aerial and face images. The strengths of the approach include:

- The use of the 3-D surface mesh allows us to deal with self-occlusions and thus effectively merge information from several potentially very different viewpoints to eliminate "blind-spots."

-- By combining stereo and shape from shading, and weighing appropriately the reliability of their respective contributions, we can obtain results that are better than those produced by either technique alone.

-- Using the facets to perform the stereo computation frees us from the constant-depth assumption that standard correlation-based stereo techniques make. It becomes possible to recover accurately the depth of sharply sloping surfaces (such as that of a sharp ridge).

-- The shape from shading component does not make a constant-albedo assumption unlike most shading algorithms. Instead, we only make the weaker and much more general assumption that albedoes vary slowly across textureless areas.

Appendix A:

An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames

International Journal of Computer Vision, 9:2, 113–136 (1992)

An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames

YVAN G. LECLERC AND MARTIN A. FISCHLER
Artificial Intelligence Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025

Received

Abstract

Line drawings provide an effective means of communication about the geometry of 3D objects. An understanding of how to duplicate the way humans interpret line drawings is extremely important in enabling man–machine communication with respect to images, diagrams, and spatial constructs. In particular, such an understanding could be used to provide the human with the capability to create a line-drawing sketch of a polyhedral object that the machine can automatically convert into the intended 3D model.

A recently published paper (Marill 1991) presented a simple optimization procedure supposedly able to duplicate human judgment in recovering the 3D "wire frame" geometry of objects depicted in line drawings. Marill provided some impressive examples, but no theoretical justification for his approach. Here, we introduce our own work by first critically examining Marill's algorithm. We provide an explanation for why Marill's algorithm was able to perform as well as it did on the examples he presented, discuss its weaknesses, and show very simple examples where it fails. We then provide an algorithm that improves on Marill's results. In particular, we show that an effective objective function must favor both symmetry and planarity—Marill deals only with the symmetry issue. By modifying Marill's objective function to explicitly favor planar-faced solutions, and by using a more competent optimization technique, we were able to demonstrate significantly improved performance in all of the examples Marill provided and those additional ones we constructed ourselves. Finally, we examine some questions relevant to the implications of this work for understanding the human ability to interpret line drawings.

1 Introduction

The interpretation of line drawings has been an important focus for research in machine vision since the field's inception. There seems to be little question that human subjects can easily recover 3D models from 2D line drawings depicting many classes of objects. One such class of special interest has been called the "blocks world." This class consists primarily of polyhedral solids in 3D Euclidean space and the projections of the visible edges of these objects onto a 2D plane (which we call the line drawing). Given a single line drawing of a blocks world scene, normal human subjects will usually arrive at the same 3D interpretation, even though there may be a very large number of possible 3D objects that could have produced the given drawing.

Beginning with the work of Guzman in 1968, there has been a concerted effort by vision researchers to develop an algorithmic procedure that could duplicate human performance in interpreting line drawings, at least with respect to blocks world objects. A significant body of work in this area was produced by such prominent scientists as Clowes (1971), Huffman (1971), Waltz (1972), Mackworth (1973), Kanade (1980), Draper (1981), and Sugihara (1982, 1984). However, the problem as originally formulated, devising a procedure for recovering psychologically plausible 3D models from line drawings, remains unsolved. (A psychologically plausible reconstruction of a line drawing is the one that virtually all people will accept.)

The earliest work by Guzman was heuristic in nature, failed in many cases where humans had no trouble in finding appropriate interpretations, and did not actually return a 3D model, but rather partitioned the scene into separate polyhedral objects. Clowes, Huffman, Waltz, Mackworth, and Kanade formalized and

extended the work of Guzman, but did not solve the original problem. They were (usually) able to label the edges of the line drawing to correctly reflect a consistent 3D interpretation if one existed, or could assert that the drawing did not correspond to a realizable blocks world scene. Mackworth and Kanade explicitly exploited the planarity of the faces of blocks world and "Origami" objects (by employing a "gradient space" representation) to accomplish a form of semiquantitative recovery. In addition to consistent edge labeling, they could also constrain the relative orientation of the faces of the target 3D model. The labels could describe the edges as being convex, concave, occluding, and so forth, but still, for the general case, no explicit 3D model was returned (without introducing additional constraints) and the algorithms would make occasional errors.[1]

In a series of papers, Sugihara reformulated the realizability and recovery problems for line drawings of polyhedra (both with and without hidden lines removed) in purely algebraic terms. He required as input a specification of the vertexes defining each of the individual planar faces of the polyhedra, and also required that the implied line drawing be a general-position projection of the polyhedra. With this approach he succeeded in providing an algebraic criterion as a necessary and sufficient condition for a line drawing to represent a physically realizable polyhedral object. He could also constrain the space of feasible solutions, and obtain a unique solution if enough additional constraints were provided. These additional constraints were obtained from information beyond that provided by the line drawing (e.g., shading or texture information). Sugihara's work was an important advance, but again it fell short of the original goal. It will rarely be the case that a unique reconstruction is implied by the line drawing, and thus the primary objective of duplicating human performance in this regard is not met.[2]

Our motivation for writing this article was supplied, in part, by a recent publication authored by T. Marill (1991). He refocused on the original problem of human interpretation of single line drawings as 3D structures; he did not restrict his universe to blocks world objects nor did he demand that the line drawings be complete. The surprising thing about his work was that he used an optimization approach involving (seemingly) an almost trivial objective function, and the simplest possible descent algorithm to find a solution, and yet provided examples of reconstructed objects that were, intuitively, extremely good. (Figure 1, examples A through I, shows the line drawings used in Marill's experiments.) However, his paper provided no justification for why the algorithm should work, and thus no basis for judging its generality or insight into how it could be improved (should this be desirable).

The first reference we have found that presents the case for choosing between various interpretations of a line drawing based on an objective function is Hochberg and McAlister (1953). In their paper, they "showed that: (1) some variants of the Necker cube are more likely to be described as 2D figures, and some are more likely to be described as 3D; and (2) these differences could be predicted by an objective and plausible coding scheme. Within this scheme, the economy of description was assessed by (among other measures) the number of lines and angles contained within the coding. Thus, the costs and benefits of 2- versus 3-D interpretations could be assessed. Figures that could be coded more simply under a depth interpretation were, in fact, seen in depth; those that could not be simplified in this way were seen to lie in the picture plane" (Pomerantz & Kubovy 1981, pp. 439–440).

Barrow and Tenenbaum (1981) suggested ideas similar to Marill's for interpreting line drawings (both for simple closed curves and polyhedra), but did not pursue the ideas in greater depth. More recently, Barnard and Pentland (1983) and Pentland and Kuo (1990) have pursued Barrow and Tenenbaum's approach for simple curves and line drawings of surfaces by finding the smoothest curve (or surface) corresponding to the line drawing.

In this article we introduce our own work by first critically examining Marill's algorithm. We provide an explanation for why Marill's algorithm was able to perform as well as it did on the examples he presented, discuss its weaknesses, and show very simple examples where it fails (figure 1, examples J through N). We then provide an algorithm that improves on Marill's results for all nine of his examples, and also successfully deals with the simple cases where Marill fails. Finally, we examine some questions relevant to the implications of this work for understanding the human ability to interpret line drawings.

We see the work described here as being of both theoretical and practical interest. The practical utility of this work is its relevance to man–machine communication about 3D structures via line drawings—in particular, providing the human with the capability to create a line-drawing sketch of a polyhedral object that the machine can automatically convert into the intended

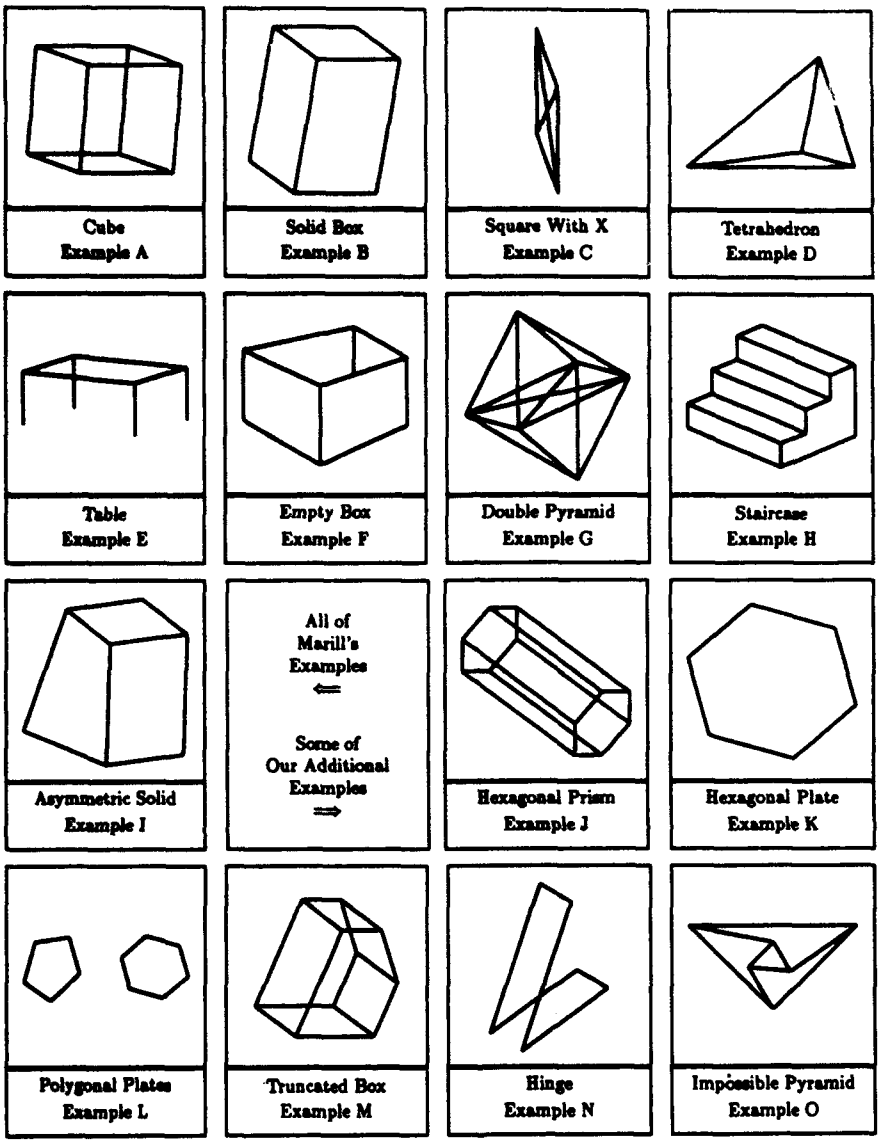


Fig. 1. The line drawings examined in this article. Examples A through I are taken from Marill's paper. Examples J through N are line drawings introduced here for which Marills' algorithm failed to recover a phychologically plausible 3D model. Example O is a line drawing for which a psychologically plausible 3D model is not feasible.

3D model. Deficiencies in providing a complete theory are not fatal, since auxiliary information can always be supplied interactively to resolve ambiguities, but the underlying theory should reduce this "side communication" to a minimum.

2 Marill's MSDA Algorithm

Marill's algorithm consists of two components, an objective function and a simple descent optimization procedure for finding a local minimum of this objective function. The objective function is simply the standard deviation of all of the angles (SDA) in the recovered 3D object with respect to their common mean. Marill calls the minimization of the SDA the MSDA principle.

The input line drawing is specified as a set of points (vertexes) and lines; each point is represented by an (x, y) coordinate pair, and each line is represented by an integer pair corresponding to the sequence numbers of the two points it joins. The representation of the recovered 3D object involves supplying a third (z) coordinate for each of the originally specified points. This is what we call the orthographic extension of the line drawing.[3] It is actually a wire frame rather than a solid object.

To evaluate the objective function for a given proposed solution, every pair of lines terminating on a

point (as defined in the input specification) is considered to form a separate angle. Thus, if five lines terminate on the same point, every potential 3D solution contains ten angles at this point that contribute to the objective function. Note that the intersection, between two lines that happen to cross at *intermediate* points of their extent in the line drawing, is not treated as a vertex, and does not contribute to the objective function (even if the lines were to lie in the same plane in the 3D reconstruction). Similarly, two *distinct* vertexes can have the same (x, y) coordinates in the line drawing, but then the line segments terminating on the distinct vertexes do not interact to form angles (even if the vertexes coincide in the 3D reconstruction).

Thus, given a line drawing with n vertexes, each possible orthographic extension is represented as a z vector having n components; the corresponding angles and SDA are computed to evaluate the proposed solution. Marill uses a descent technique to search for a best answer, recognizing that this is simply a heuristic and that this approach will find only a single local minimum of his objective function. The input object has all of its z values initially set to zero; that is, it is a flat object lying in the (x, y) plane. At each stage in the search, the SDA of the current z vector is computed and the program then looks at the children of the current vector. These $2n$ children are all of the vectors one step size away from the current vector, and are formed by both adding and subtracting a specified value (Δz) to each of the n components in the current z vector. The value of the SDA is computed for each of these $2n$ children, and the child with the minimum SDA is selected as the new current vector. This process is repeated until no improvement in the SDA is obtained, and the resulting z vector is returned as the solution for the first of three rounds of descent. Each additional round uses a smaller Δz and begins with the result of the preceding round. Marill experimentally found effective values of Δz for his three rounds to be 1, 0.5, and 0.1.

Figure 2 shows a line drawing, its internal representation as described above, and the reconstructions using Marill's algorithm and the algorithm we describe in section 3.

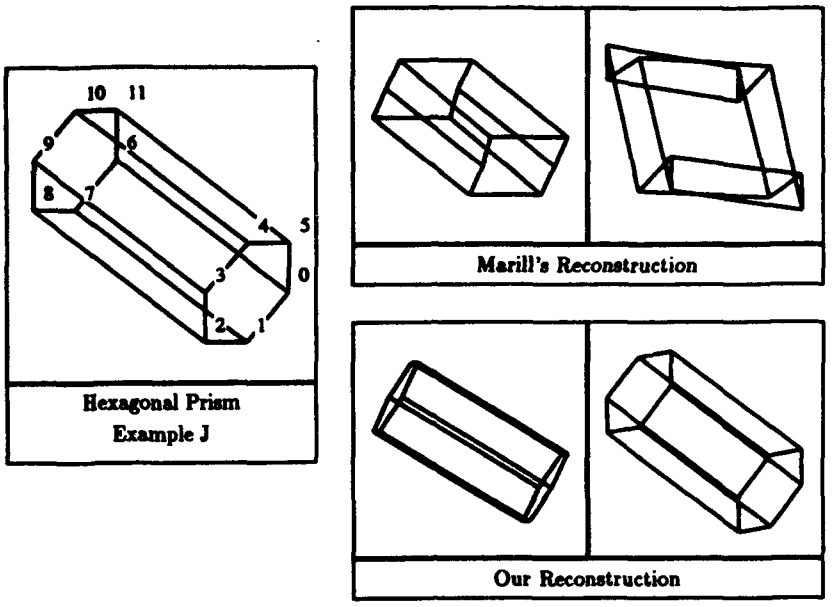
In the top left window of the figure is the input line drawing (with the vertexes numbered for reference by the written representation below). The four windows on the top right show two views of Marill's reconstruction and two views of our reconstruction. In the middle of the figure is a table showing the internal representation of the input line drawing. In the first row are the (x, y) coordinates of the vertexes, in the order shown on the drawing.[4] In the second row are the integer pairs representing the lines in the drawing. In the third row are the sequences of vertexes corresponding to the planar faces derived according to the rules of appendix A (see section 3). The reconstructions are discussed in section 3.3.

2.1 Marill's Examples

Marill described the application of his algorithm to examples A through I of figure 1. We categorize these examples along the following dimensions (based on the appearance of the input drawing and on the characteristics of the recovered 3D object):

a. —Three-dimensional [A B D E F G H I]
   —Flat [C]
b. —Blocks world (planar-faced solids with occluded edges not rendered) [B H I]
   —Origami (planar-faced, possibly hollow) [C F]
   —Wire frame of blocks world object (all edges of a blocks world object are given, and additional lines between vertexes of a planar face may be added) [A D G]
   —Restricted wire frame (every closed circuit of lines, without interior lines in the given input representation, corresponds to a planar face) [E]
   —Nonplanar wire frame (none of the above)
c. —Symmetric [A B C E G H]
   —Asymmetric [D F I]
d. —All angles (approximately) equal [A B E F H]
   —A few distinct but mostly repeated angles (C G I]
   —Mostly unequal angles [D]

For the purposes of our discussion, we use Marill's categorization and augment it with our own subjective evaluation where we disagree or need to add additional attributes to those Marill provides. It is important to remember that Marill always returns a wire frame as his solution, regardless of the categorization of the object. Thus, we would call the wire frame of a blocks world object a correct solution if it was a geometrically correct representation of the 3D geometry of the edges of the psychologically plausible blocks world object whose orthographic projection corresponded to the input line drawing, even though the wire frame does not provide an explicit representation of the grouping of lines into faces, and so forth.



Points	(1.97 -1.00) (1.32 -1.75) (0.67 -1.75) (0.68 -1.00) (1.34 -0.25) (1.98 -0.25) (-0.68 1.00) (-1.34 0.25) (-1.98 0.25) (-1.97 1.00) (-1.32 1.75) (-0.67 1.75)
Lines	(0 1) (1 2) (2 3) (3 4) (4 5) (5 0) (6 7) (7 8) (8 9) (9 10) (10 11) (11 6) (0 6) (1 7) (2 8) (3 9) (4 10) (5 11)
Faces	(0 1 7 6)(1 2 8 7)(2 3 9 8)(3 4 10 9)(4 5 11 10)(5 0 6 11)(0 1 2 3 4 5)(6 7 8 9 10 11)

	Zs	Lengths	Angles (Mean / Range)	SDA^2	DP
Original Object	0.00 0.10 0.87 1.53 1.43 0.66 -2.23 -2.12 -1.36 -0.69 -0.80 -1.57	1.0 to 4.0	100.0 90.0 to 120.0	0.060923	0.000000
Marill's Reconstruction	0.00 0.46 -2.15 -1.48 -2.19 0.72 -0.37 0.33 -2.61 -1.92 -2.36 0.31	1.0 to 3.4	84.0 47.5 to 111.2	0.110660	0.044710
Our Reconstruction	0.00 0.12 0.96 1.66 1.55 0.71 -1.99 -1.87 -1.04 -0.35 -0.47 -1.31	1.0 to 3.9	100.0 88.6 to 122.6	0.061289	0.000000

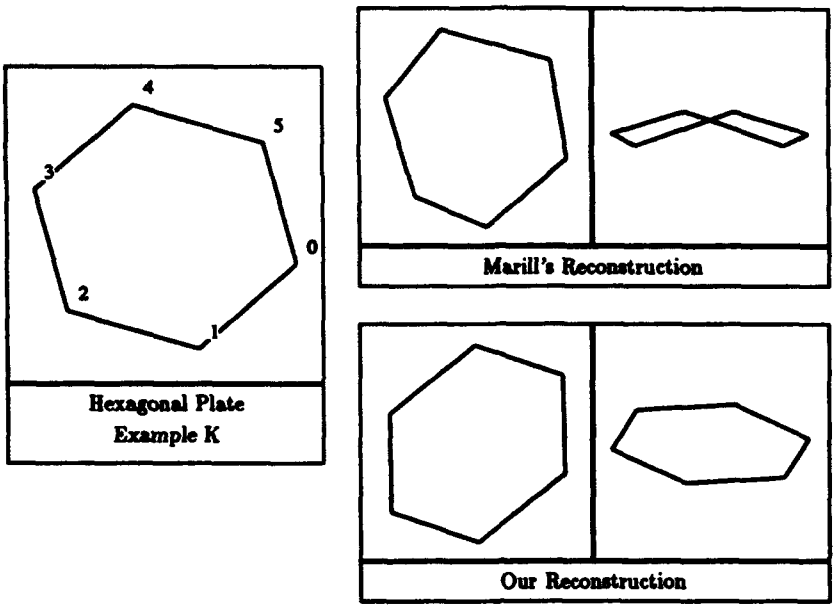
Fig. 2. Example J. This line drawing was created by orthographically projecting a specific 3D wire frame object. In this case, the object was a regular hexagonal prism. Although arbitrary line drawings can be used as input to the reconstruction algorithms described in this article (with greater or lesser success in reconstruction), all of the examples introduced here were created by starting with specific 3D objects. The panels in the upper right show two views of the object reconstructed by Marills' algorithm. The first view is of the object rotated about the vertical axis by 30 degrees, and the second is of the object rotated about the horizontal axis by 90 degrees. The two panels in the lower right show two views of the object reconstructed by our algorithm. The table below this is the internal representation of the line drawing used by the reconstruction algorithms. Note that intersections such as those between lines (1 7) and (2 3) are not represented. Marill's algorithm uses only the first two components of this representation. The third component (faces) is derived from the line drawing using the algorithm described in section 3.1. The table at the bottom shows the results of the reconstructions in written form.

Examples A, B, E, F, and H can all be visualized as approximately equiangular three-dimensional objects. That is each of the objects has an equiangular 3D wire frame as a psychologically plausible solution. Since these equiangular solutions exactly satisfy Marill's minimum standard deviation of angles (MSDA) criterion, it is obvious why Marill's objective function should prefer what we accept as the correct solutions in these cases. In the other four cases, supposedly representative examples of the ability of Marill's algorithm to deal with complicated structures having unequal angles, reasonably correct solutions are also recovered, and it is this performance we wish to understand.

2.2 *The Performance of the MSDA Principle*

Given its overall simplicity, it would be quite remarkable if the MSDA principle generally converged to a psychologically plausible reconstruction. Unfortunately, it is rather easy to find examples where this is not the case, contrary to Marill's implied competence for the principle.

Examples J through N of figure 1 are line drawings for which Marill's algorithm converged to solutions that are clearly psychologically implausible, even though these drawings are not significantly more complicated or more asymmetric than the examples that Marill used



Points	(0.96 -0.27) (0.24 -0.89) (-0.72 -0.61) (-0.96 0.27) (-0.24 0.89) (0.72 0.61)
Lines	(0 1) (1 2) (2 3) (3 4) (4 5) (5 0)
Faces	(0 1 2 3 4 5)

	Zs	Lengths	Angles (Mean / Range)	SDA^2	DP
Original Object	0.00 0.32 0.24 -0.15 -0.48 -0.40	1.0 to 1.0	120.0 120.0 to 120.0	0.000000	0.000000
Marill's Reconstruction	0.00 0.22 -0.12 0.00 0.22 -0.12	0.9 to 1.1	116.2 116.2 to 116.2	0.000000	0.030363
Our Reconstruction	0.00 0.34 0.28 -0.11 -0.43 -0.38	1.0 to 1.0	120.0 119.6 to 120.4	0.000029	0.000000

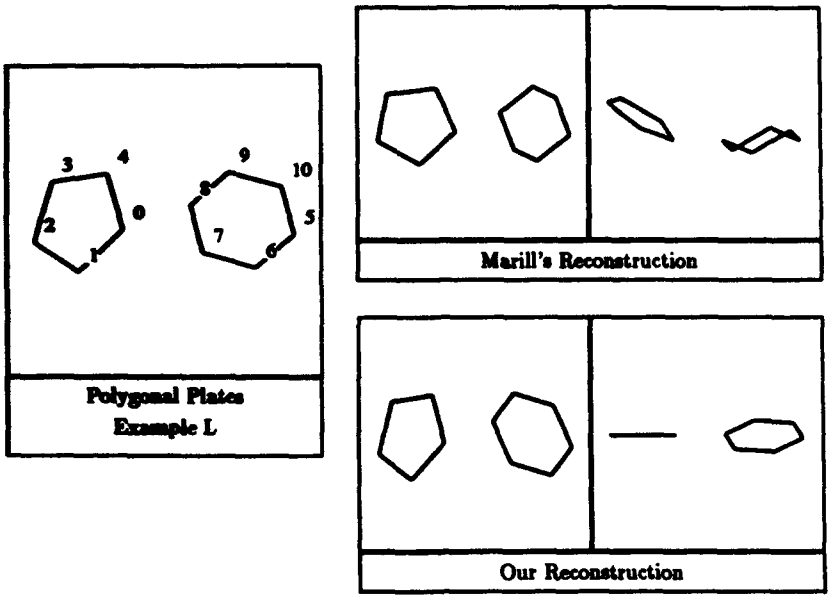
Fig. 3. Example K.

(figures 2, 3, 4, 5, and 6 illustrate both Marill's reconstructions and our reconstructions, as described in section 3). In Examples J and K it would appear that the fault could lie with Marill's use of a descent algorithm because the SDA of the psychologically plausible answer is less than or equal to the SDA for the solution Marill actually obtains. Thus, one can argue that a more competent global search strategy could have found the psychologically plausible answer using the same objective function. However, Examples L, M, and N are line drawings for which the SDA of Marill's solution is significantly lower than that of the psychologically plausible solution. Thus, the MSDA principle is clearly not adequate to *reliably* handle even simple line drawings.

Before discussing ways of augmenting the MSDA principle to obtain a more competent principle and algorithm, we attempt to explain the performance of MSDA for line drawings depicting objects that are not equiangular.

2.3 Evaluating the Performance of the MSDA Principle

It is not immediately obvious why the MSDA principle should prefer a psychologically plausible answer if the object depicted in the line drawing contains two or more significantly different angles (e.g., C, D, G, I, and J). Marill offers no explanation for this phenomenon, and thus no way to judge the conditions under which his algorithm should be expected to succeed or fail. In this section we provide a partial explanation for cases (such as C, G, J, K, and L) that have critically important attributes—the psychologically plausible reconstruction is a 3D *planar-faced* object whose faces are either equiangular or form "complete-star" configurations (see appendix B).



Points	(0.81 -0.19) (-0.02 -0.96) (-0.82 -0.41) (-0.49 0.71) (0.52 0.85) (3.96 -0.27) (3.24 -0.89) (2.28 -0.61) (2.04 0.27) (2.76 0.89) (3.72 0.61)
Lines	(0 1) (1 2) (2 3) (3 4) (4 0) (5 6) (6 7) (7 8) (8 9) (9 10) (10 5)
Faces	(0 1 2 3 4)(5 6 7 8 9 10)

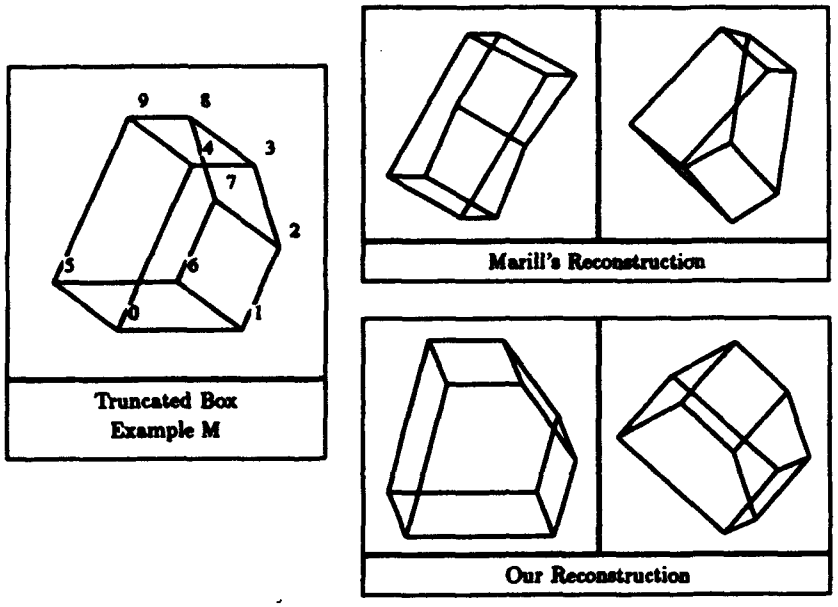
	Zs	Lengths	Angles (Mean / Range)	SDA^2	DP
Original Object	0.00 0.29 0.94 1.06 0.47 0.47 0.15 0.23 0.63 0.95 0.87	1.0 to 1.2	114.5 108.0 to 120.0	0.010876	0.000000
Marill's Reconstruction	0.00 0.30 0.93 1.04 0.46 0.00 0.31 -0.24 0.00 -0.31 0.24	0.9 to 1.2	108.0 107.7 to 108.2	0.000005	0.165552
Our Reconstruction	0.00 0.00 0.00 0.00 0.00 0.08 0.41 0.35 -0.04 -0.36 -0.30	1.0 to 1.2	114.5 97.9 to 120.4	0.018157	0.000000

Fig. 4. Example L. Note that Marill's unacceptable reconstruction has an SDA that is significantly lower than that of the psychologically plausible original object. Thus, the MSDA principle itself has failed in this instance.

To establish the role played by the above geometric attributes, we define the *planar orthographic extension of a simple closed 2D circuit* in a line drawing to be any orthographic extension for which the corresponding 3D contour is planar. If a line drawing contains more than one simple closed 2D circuit, then a *planar orthographic extension of the entire line drawing* exists if we can cover the line drawing with a set of simple closed 2D circuits such that (a) every angle in the drawing is included in at least one circuit, and (b) each circuit projects to a 3D planar contour.[5]

In appendixes B, C, and D, we provide a number of theorems that are pertinent to understanding the effectiveness of the MSDA principle applied to planar orthographic extensions. The main theorem, appendix D, asserts that solutions with certain symmetries correspond to the global minimum of the SDA over all planar orthographic extensions (the specific symmetry condition we examine is that all faces must either be equiangular or form complete-star configurations).

Consequently, if there were some way to consider as possible solutions only the *planar* orthographic extensions of a line drawing (such as the psychologically plausible solutions for examples A, B, C, G, J, K, and L), these solutions would be global minima of the SDA because of the angular symmetry they exhibit. We show in example L that Marill's algorithm is not constrained to search only for planar solutions; while it will also find solutions with nonplanar faces that have lower SDAs then the planar solutions, there is still the possibility that MSDA shows at least a weak inherent preference for planarity. While we cannot completely rule out this possibility, it appears that the geometric constraints inherent in the specific examples Marill selected, rather than MSDA itself, are largely responsible for finding planar-faced solutions. Specifically, triangles in the line drawing will always produce planar faces in the orthographic extension, and as we prove



Points	(0.15 -0.06) (0.80 -0.06) (0.99 0.38) (0.86 0.81) (0.54 0.81) (-0.18 0.19) (0.46 0.19) (0.66 0.63) (0.53 1.06) (0.20 1.06)
Lines	(0 1) (1 2) (2 3) (3 4) (4 0) (5 6) (6 7) (7 8) (8 9) (9 5) (0 5) (1 6) (2 7) (3 8) (4 9)
Faces	(0 5 9 4)(1 0 5 6)(2 1 6 7)(3 2 7 8)(4 0 5 9)(4 3 8 9)(0 1 2 3 4)(5 6 7 8 9)

	Zs	Lengths	Angles (Mean / Range)	SDA^2	DP
Original Object	0.00 0.77 0.61 0.06 -0.32 0.28 1.04 0.88 0.34 -0.04	0.5 to 1.0	96.0 90.0 to 135.0	0.071281	0.000000
Marill's Reconstruction	0.00 0.68 0.67 -0.17 -0.37 0.31 0.97 0.93 0.18 0.01	0.4 to 1.0	95.4 73.5 to 125.0	0.047822	0.004897
Our Reconstruction	0.00 0.57 0.49 0.10 -0.20 0.37 0.94 0.86 0.46 0.16	0.4 to 1.0	96.0 80.7 to 132.2	0.059677	0.000000

Fig. 5. Example M. Note that our reconstruction has a slightly lower SDA than that of the original object, indicating the preference of our algorithm for equiangular faces.

in appendix B, a closed four-sided polygonal space curve with 90-degree angles at each vertex will always be a planar configuration. Since in Marill's examples listed above, all the faces satisfy these two geometric conditions, we see why both the desired planarity and symmetry are present in the computed solutions.[6]

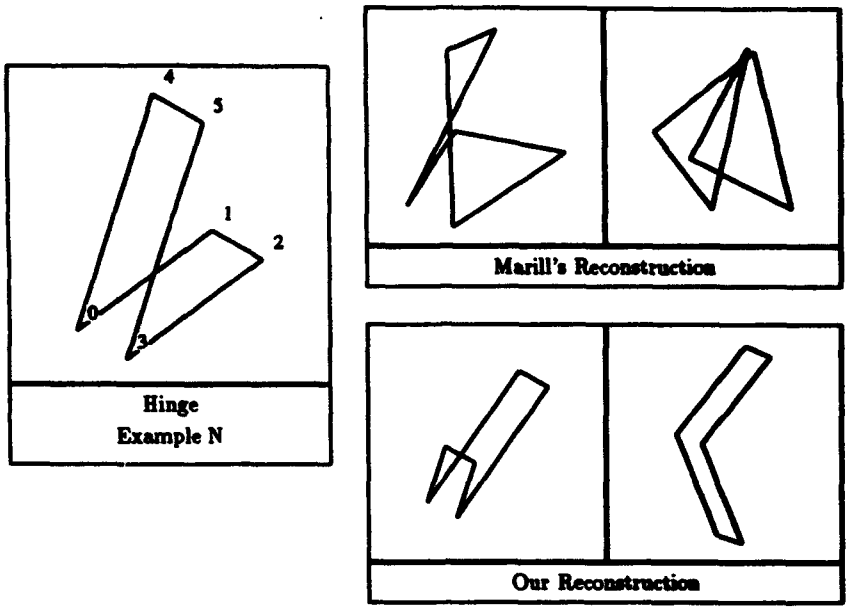
Marill offers only two examples (D and I) that are not clear instances of the above analysis (all angles equal, or symmetric planar faces). His solution for example I is at least questionable since it does not recover the wire frame of a polyhedral solid (our algorithm finds such a solution; there is a further discussion of this subject in sections 3.3 and 4). However, this solution has almost all of its angles equal to 90 degrees, and so it needs no further explanation if we accept it as correct.

Marill's solution to the asymmetric drawing of example D looks very reasonable; it has all its angles fairly well distributed between 40 and 70 degrees, and we have not found a more symmetric (equiangular) orthographic extension for this line drawing. However, because the input line drawing is a completely connected set of triangular faces, all solutions are constrained to have planar faces. Thus, a large range of psychologically plausible objects is accessible to any reasonable algorithm.

In summary, there is an understandable reason why Marill's MSDA principle will sometimes tend to select planar symmetric 3D wire frames when a purely equiangular solution is not possible. But we also see that MSDA will make unacceptable errors, even in simple cases, because it is not constrained to prefer solutions with planar faces unless the geometry of the line drawing itself forces planarity.

3 Our Planarity Enforcing MSDA Algorithm

What's missing in the MSDA principle is a means for enforcing the planarity of specified faces. There are two



Points	(-0.58 0.24) (0.95 1.36) (1.50 1.04) (-0.02 -0.08) (0.30 2.89) (0.86 2.56)
Lines	(0 1) (1 2) (2 3) (3 5) (5 4) (4 0)
Faces	(0 1 2 3)(0 4 5 3)

	Zs	Lengths	Angles (Mean / Range)	SDA^2	DP
Original Object	0.00 0.64 -0.12 -0.77 -0.47 -1.24	1.0 to 2.8	75.0 45.0 to 90.0	0.137078	0.000000
Marill's Reconstruction	0.00 1.21 -1.17 -0.41 -1.18 1.26	2.0 to 3.3	63.8 62.7 to 65.0	0.000283	0.196270
Our Reconstruction	0.00 1.93 1.69 -0.21 -2.21 -2.40	0.7 to 3.6	89.8 88.8 to 91.3	0.000291	0.000000

Fig. 6. Example N. The SDA of Marill's unacceptable reconstruction is again significantly lower than that of the psychologically plausible original object.

parts to this problem: (1) finding those faces in the line drawing that should be planar in the 3D reconstruction, (2) and enforcing the planarity of these faces during, or at least by the end of, the optimization process.

3.1 Finding Planar Faces

The following algorithm for finding the planar faces is based on a set of psychological assumptions presented in appendix A. The requirements of items 3, 4, and 5 from appendix A have been composed into the following algorithm. (In the following discussion, we define a face in the line drawing to be a sequence of vertexes.)

First, all simple (nonself-intersecting) closed circuits containing more than three lines are found. (Triangles are necessarily planar, so they need not be considered.) Those circuits that are either: (1) completely empty of both lines and vertexes (such as the faces of example B); or (2) both convex (in the line drawing) and free of internal circuits (such as all the faces of example J) are considered to be planar faces of the wire frame; call this initial set $\mathcal{P}_0$. A circuit is defined to be an internal circuit to a convex circuit if: (1) all of its vertexes lie within the convex circuit; and (2) it terminates in two nonadjacent vertexes of the convex circuit.

Added to $\mathcal{P}_0$ are those circuits, defined by the following algorithm, that are not subsets of any circuit in $\mathcal{P}_0$. First, all triples of consecutive lines such that the first and third lines are parallel are found (the two planar faces of example N fall into this category, as do the "table legs" of example E). Then, if possible, each triple of lines is extended with additional consecutive lines such that all even-numbered lines are parallel to

each other and all odd-numbered lines are parallel to each other. An example of a closed circuit found this way is the side of the staircase facing the viewer in example H; the side of the staircase opposite the viewer is an example of an open circuit found using this same rule.

Finally, pairs of parallel lines lie on planar faces in general position, so the four vertexes of the pair of lines are defined to form a planar face (whether or not the vertexes are connected by lines in the line drawing). If the pair of lines are not already a subset of a previously found planar face, these are added to $\mathcal{P}_0$.[7]

The above procedure is remarkably robust in dealing with unconstrained line drawings. For example, we have yet to find a case where this procedure proposes a psychologically implausible planar face (it even found faces in our test cases that we had not originally recognized as being planar—such as the back side of the staircase in example H). However, it will sometimes miss finding a concave planar face leaving the 3D model underconstrained, and this can result in the reconstruction of a psychologically implausible 3D wire frame. If we know that the line-drawings to be processed are restricted to the projections of blocks world objects with all planar intersections included in the drawing (i.e., no hidden lines removed), then we can be assured that no faces are missing by (omitting some details here) first employing the above procedure, next removing all lines' edges from the drawing that are assigned to two faces, and then repeating this whole process on the reduced line drawing until all the edges have been assigned to exactly two faces (there are some special-position configurations in which three or more faces have a single edge in common that we presently do not deal with). For this more constrained universe of line-drawings where we correctly and completely identify all the planar faces, we have yet to encounter a case where our algorithm produces a psychologically implausible 3D model.

3.2 *Enforcing Planarity*

The second requirement, enforcing planarity, is accomplished by adding a term to the objective function that is zero when all the designated planar faces are actually planar, and increases in value as the faces deviate from planarity (call this term DP). The new objective function, $E(\lambda)$, is a linear combination of the previously defined SDA term and the new DP term:[8]

$$E(\lambda) = \lambda SDA^2 + (1 - \lambda)DP$$

Note that minimizing $E(\lambda)$ favors planar faces, but strict planarity is not necessarily assured. This is not quite what we would like in the ideal case. Ideally, we would like to find the orthographic extension of the line drawing with the lowest SDA that has exactly planar faces (i.e., for which $DP = 0$).[9] To achieve this, we use a continuation method (Leclerc 1989; Witkin et al 1987), which is a sequence of descent steps applied to $E(\lambda)$, for decreasing values of λ. The sequence begins with the initial condition that Marill suggests ($z = 0$ for all points) and with some initial $\lambda_0 \leq 1$. Then, λ is decreased by a given amount and the descent algorithm is applied anew, starting at the solution found for the previous value of λ. This is repeated until λ is sufficiently close to zero so that no additional changes occur with further reductions in λ.

Why not simply start with λ close to zero in the first place? The reason is that when λ is sufficiently close to zero, the local minima of $E(\lambda)$ are determined only by the planarity component. Thus, simply starting with λ close to zero would not allow us to find solutions with low SDAs (in fact, when $\lambda = 0$, the original line drawing, which is planar, is a local minimum of $E(\lambda)$). Although we cannot affect the shape of $E(\lambda)$ when λ is small, we can choose the starting point for the descent algorithm. Thus, the purpose of the continuation method is to choose a sequence of starting points that are first strongly influenced by the SDA term, but which eventually become dominated by the DP term. The method is not guaranteed to find a global minimum of the objective function, but has yielded excellent answers for all the examples discussed in this paper.

We define the deviation from planarity term, DP, as the sum of terms DP_i, where DP_i is zero when face f_i is planar, and increase as the face deviates from planarity. We have found two useful definitions of the DP_i. The first is a strong planarity term that will not allow a face to fold from one planar configuration to another planar configuration, but applies only to convex faces. To see how a face can fold from one planar configuration to another one within the context of the optimization we are performing, consider a line drawing of a square. When all of the z values of the vertexes are zero, the face is planar. By letting the z values of the first and third vertexes become arbitrarily large, the face "folds" into a configuration that, in the limit, is also planar. In order to detect and avoid this folding whenever possible, we define DP_i to be the following function ($DP1$) whenever face f_i is convex in the line drawing ($DP1$ is based on item 6 in appendix B):

Let n be the number of sides in the face, and α_j be the angle at the j^{th} vertex. Then,

$$DP1 = \left((n-2)\pi - \sum_j \alpha_j \right)^2$$

A weaker measure of planarity, $DP2$, applicable to all faces, is based on the observation that the normals defined by pairs of consecutive pairs of lines should lie in the same direction (this is analogous to the notion of torsion for a curve):

$$DP2 = \sum_j \left[1 - \left(\frac{(l_{j-1} \times l_j) \cdot (l_j \times l_{j+1})}{\|l_{j-1} \times l_j\| \, \|l_j \times l_{j+1}\|} \right) \right]^2$$

where l_j is the j^{th} line of planar face f_i and $j-1$ and $j+1$ refer to the previous and next lines in the face, respectively (i.e., the subscripts are taken modulo the number of lines in the face).

The combined DP term is the sum of: (1) the sum of $DP1$ over all convex faces, and (2) the sum of $DP2$ over all nonconvex faces divided by the number of angles in all of the nonconvex faces.

3.3 Results

Figures 2 through 6 illustrate the results of our *planarity enforcing MSDA* algorithm, and allows one to compare them with both Marill's reconstructions and the original 3D objects that were used to generate the line drawings. The "original 3D objects" presented in our figures are the psychologically plausible solutions that we expect the program to recover. We started with actual 3D wire frames, rather than arbitrary line drawings as an experimental expedient, since most random line drawings will not induce the perception of a 3D configuration in human subjects.

The reconstructions are illustrated both graphically (as two views in the upper third of each figure) and in tabular form in the lower third. The first column of the table lists the z coordinates of each object, the second column is the range of lengths of the lines of each object, the third column is the mean and range of the angles formed by all line pairs meeting at a common vertex, the fourth column is the standard deviation of angles (SDA) of each object, and the fifth column is the deviation from planarity (DP) of each object. To simplify the comparison of the results, the recovered z coordinates have been normalized so that the first point always has $z = 0$, and the second coordinate is always positive (this normalization procedure has no effect on the objective function).

We also applied our algorithm to examples A through I from Marill's paper. Since his algorithm produced approximately planar-faced solutions by itself in all cases but example I, it isn't surprising that our algorithm produced solutions almost identical to his. The greatest deviation from his result was for example I, because Marill's algorithm recovered a significantly nonplanar face for the leftmost face of the line drawing.

In all of the examples, the Δzs we used for Marill's algorithm (both as a stand-alone algorithm and within the continuation method) were 0.125, 0.0625, 0.03125, 0.015, and 0.007. We used a smaller initial Δz than Marill suggests because the larger one often forced the algorithm out of the valley of attraction of the current local minimum. Decreasing Δz by a factor of two generally allowed the algorithm to run in the fewest number of iterations. Using a smaller final Δz allowed the algorithm to produce significantly more accurate solutions. In the continuation method, λ was started at 0.25, and was decreased by a factor of two a total of ten times.

Example J (figure 2) illustrates Marill's reconstruction for a line drawing of a rectangular hexagonal prism. This reconstruction not only appears psychologically implausible from these two views, but, as we discuss in the following section, the reconstructed object does not appear rigid when rotated in real time. It would appear that at least part of the reason for this result is that the recovered faces are clearly nonplanar, as shown by the value of DP in the table. The reconstruction obtained by using the planarity enforcing MSDA algorithm is almost identical to the original hexagonal prism.

In example K, we see that the MSDA principle is ambiguous for simple line drawings. Marill's reconstruction takes the line drawing of a planar hexagonal plate ($SDA = 0.0$) and reconstructs a nonplanar object, also with SDA = 0.0. By enforcing planarity, however, our reconstruction is quite close to the original hexagonal plate.

In examples L and N, we see further evidence that the MSDA principle by itself is inadequate for even simple line drawings. In both examples, Marill's reconstruction has a significantly lower SDA than the original object, and we consider both of these reconstructions to be psychologically implausible. Our reconstruction of example L is quite close to the original object, modulo an additive constant and flip of the z coordinates of the second object (which is invisible to the objective

function). Example N is a fairly ambiguous figure, and our reconstruction favored a "hinge" with all angles close to 90 degrees (the original object had a "hinge-angle" of 45 degrees). Because of the ambiguity of the figure, there exists a family of reconstructions that we consider psychologically plausible, including ours.

Example M shows the reconstruction of a figure for which some of the planar faces are not equiangular. Again, because some of the faces had more than four sides, Marill's algorithm failed to recover a psychologically plausible object. Our reconstruction is reasonably good, but it did adjust the right angles in the large face by as much as 13 degrees in order to make the angles in that face closer to being equal. Nonetheless, we consider the reconstruction to be psychologically plausible.

3.4 Stability and Robustness of the Planarity Enforcing MSDA Algorithm

We have examined the stability and robustness of our algorithm in two ways. The first was to examine the behavior of the algorithm applied to different projections of the same 3D objects, but always using the same initial conditions for the optimization, namely $z = 0$ for all vertexes. The second was to examine the behavior of the algorithm for different initial conditions.

We ran the planarity enforcing MSDA algorithm on at least 32 randomly chosen projections of the 3D objects used to create the line drawings of examples A through N.[10] For virtually every projection of each of these objects, the algorithm reconstructed the object as well as it did for the original projection. For example, figure 7

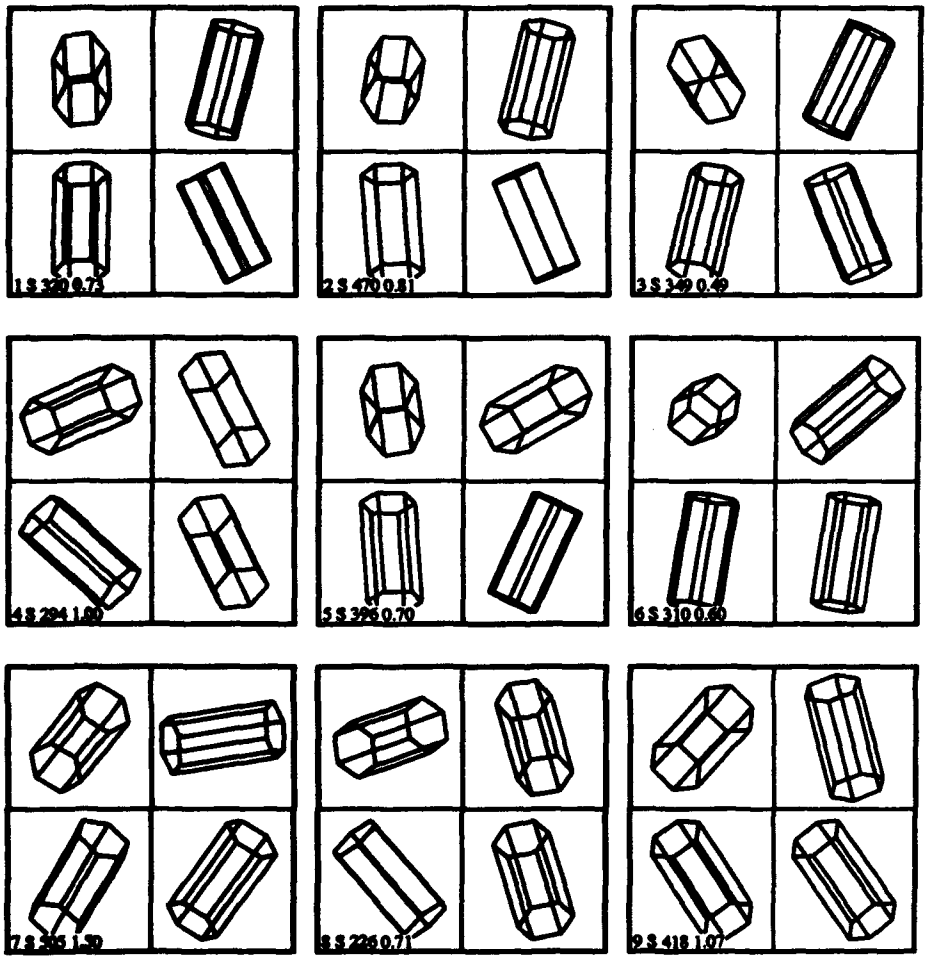


Fig. 7. Nine projections of the hexagonal prism, and our corresponding reconstructions. The projections used as original line drawings are shown in the lower left-hand corner of each group of four. The original line drawing is annoted by: (1) the projection number; (2) the letter S when the planar faces found for that line drawing were the same as for the original projection, and D otherwise; (3) the number of iterations required for convergence; and (4) the largest difference between corresponding angles in the reconstruction and the original object, in degrees. The other three line drawings are three views of the reconstruction.

shows nine projections and the corresponding reconstructions for the hexagonal prism (Marill's algorithm failed for all of these projections). An example of a near failure is shown in figure 8, where the eighth projection of the staircase is almost in special position, producing the largest error, and using the greatest number of iterations. In fact, when the rule adding all pairs of parallel lines as planar faces is removed, the algorithm leaves the z values virtually unchanged from their initial values (not illustrated here). In summary, in approximately 500 trials, either the planarity enforcing MSDA algorithm correctly reconstructed the original object, or it left the line drawing as an "uninterpreted" flat object.

By comparison, the MSDA algorithm is relatively unstable, even for the line drawings one might expect it to get right. For example, figure 9 shows nine projections and the corresponding reconstructions using the MSDA algorithm, for a cube in which all of the angles should be exactly equal. Note that projections 1 and 9 produce psychologically implausible reconstructions.

In a second set of experiments, we used a random-number generator to provide twenty sets of initial zs in the range -1 to 1 for examples A through N.[11] With the exception of example D, which was always correctly reconstructed, the MSDA algorithm failed to converge to a psychologically plausible solution in at least four of the twenty trials on each of the other line drawings, and produced an average of ten failures per line drawing. In other words, the SDA term by itself has many local minima that descent algorithms will fall into.

On the other hand, the planarity enforcing MSDA algorithm succeeded in converging to a psychologically plausible solution in all trials but one (it failed in one trial of example N, the hinge.)[12] This extremely robust performance was somewhat unexpected. We believed that the initial condition, $z = 0$ for all vertexes was an important component of the continuation

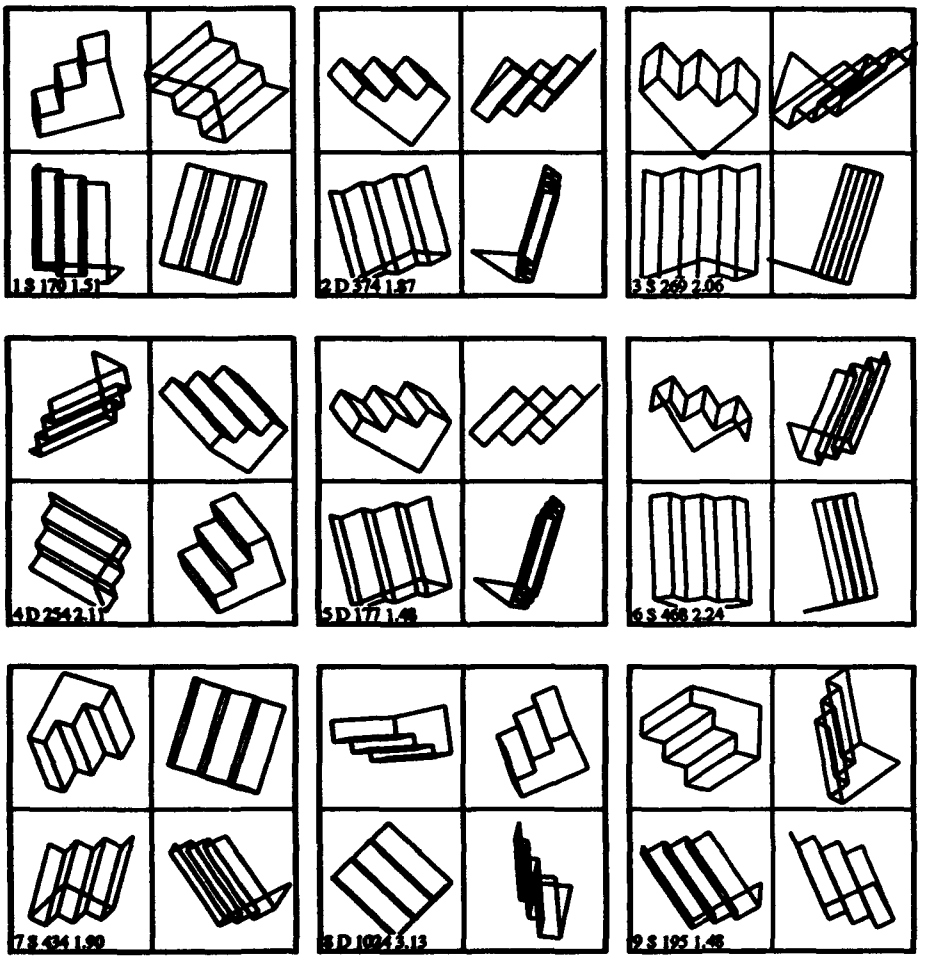


Fig. 8. Nine projections of the staircase, and our corresponding reconstructions. Note that the eighth projection is very nearly in special position, with many vertexes and lines overlapping in the line drawing. The continuation method had the largest error and used the greatest number of iterations for this case. When the rule adding all pairs of parallel lines as planar faces is removed, the continuation method prefers the original line drawing (all zs constant) as the intepretation, which is certainly psychologically plausible.

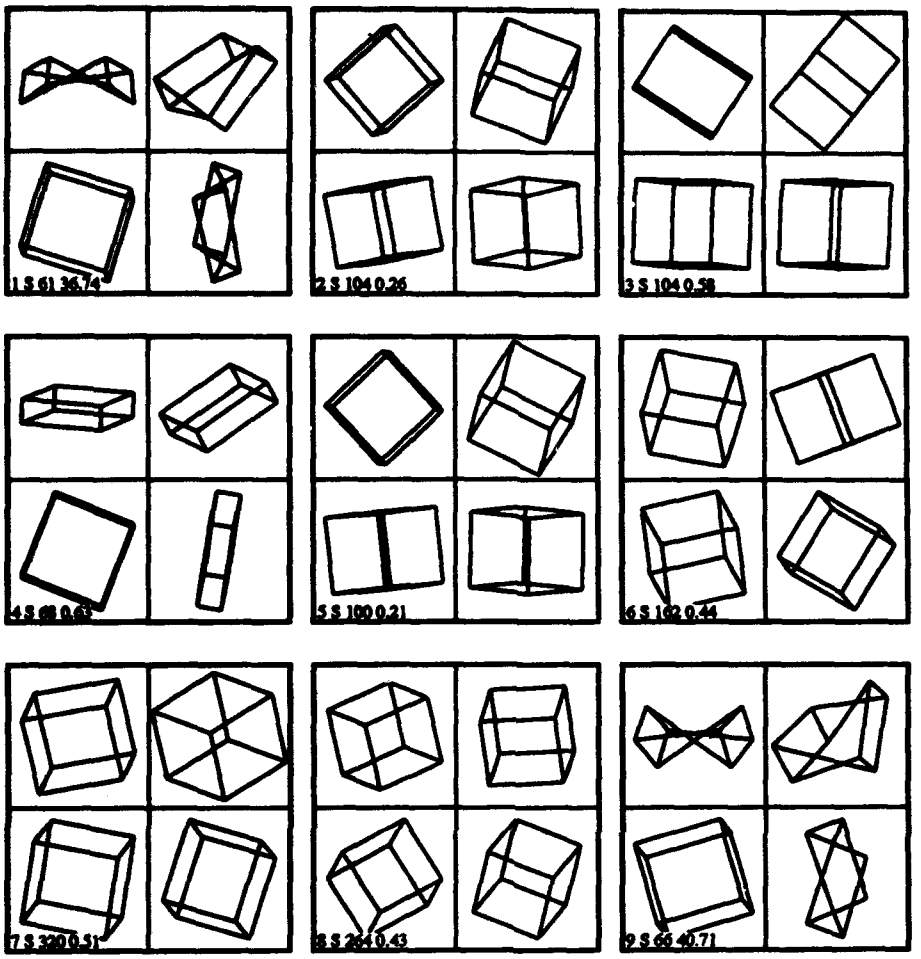


Fig. 9. Nine projections of the cube, and the corresponding reconstructions using Marill's algorithm. Note that projections 1 and 9 produced psychologically implausible reconstructions.

method. However, it would appear from the results of these experiments that the imposition of the planarity term in the continuation method severely curtails, or eliminates, psychologically implausible minima. One might conjecture that, for most line drawings, there is one[13] (or perhaps a very few) psychologically plausible local minima in the SDA when the zs are constrained to a *planar* orthographic extension.

3.5 Reconstruction Time

The specific descent algorithm defined by Marill, and described here, has the nice property that it's easy to describe and easy to implement, no matter what the objective function may be; however, it is typically quite inefficient. One of the better descent algorithms is the conjugate gradient algorithm. To estimate achievable run times, we implemented the conjugate gradient algorithm described in *Numerical Recipes* (Press et al. 1986). The algorithm requires an objective function (in this case, $E(\lambda)$) and the gradient of the objective function (in this case, a function that returns a vector whose i^{th} element is the partial derivative of $E(\lambda)$ with respect to z_i). Analytically deriving the gradient of $E(\lambda)$ is rather painful, so instead we used a simple numerical approximation; this involves evaluating the objective function for each vertex, which is expensive. A more efficient implementation that only recomputes those components of the objective function that change when a given vertex changes could reduce the following run times by a factor of four or better.

Table 1 gives the number of iterations/run time (in seconds) for three example line drawings. These experiments were run on a Symbolics 3645, so we would expect about a factor of ten improvement if algorithms

were implemented in C on a modern workstation such as a SUN SPARC-2 (according to some simple benchmarks that we ran). The last column gives the expected run time for an optimized conjugate gradient algorithm running on a SPARC-2.

Table 1. Number of iterations/run time for three examples.

Example	*Original Descent on Symbolics*	*Conjugate Gradient on Symbolics*	*Conjugate Gradient on SPARC-2*
Cube	187/199	15/15	15/0.375
Tetrahedron	46/9	14/16	14/0.4
Hexagon prism	406/1306	33/186	33/4.65

Note that the conjugate gradient algorithm improves the run-time considerably for all but the simple tetrahedron line drawing. On a SPARC-2, the run times are such that the time required to reconstruct a line drawing is small relative to the time it would take to manually enter the drawing. That is, the run times are well within "interactive time."

3.6 A Reduced Search Space Technique for Obtaining Exact Planar MSDA Reconstructions

In the planarity enforcing MSDA algorithm described in section 3.2, planarity is not strictly enforced, but rather, nonplanarity is penalized during the optimization process. This approach almost always produces faces that are very nearly planar at the end of the optimization process. There is a very efficient way to *strictly* enforce planarity during the MSDA optimization for line drawings of strictly planar-faced wire frames, described below. The problem with this approach is that if the line drawing does not actually correspond to a planar-faced wire frame, or if the line drawing is not accurate, the resulting reconstruction will typically be psychologically unacceptable—we lose the graceful degradation provided by the planarity enforcing MSDA.

The following method for strictly enforcing planarity is based on the observation that there are far fewer degrees of freedom in a planar-faced object than there are vertexes (to reemphasize, this method is only applicable to line drawings of strictly planar-faced wire frames). One way of expressing this observation is in terms of a subset of vertexes, that we call the *free vertexes*, whose z values uniquely determine the z values of all of the other *dependent* vertexes by virtue of the planarity of certain faces. For instance, given the planar faces of the hexagonal prism of figure 2, specifying the depth of the four vertexes 0, 1, 2, and 6 uniquely determines the depth of the other vertexes: the depth of vertexes 3, 4, and 5 are determined by constraining them to lie on the same planar face as vertexes 0, 1, and 2; similarly, vertex 11 is determined by vertexes 0, 5, and 6; vertex 10 by vertexes 4, 5, and 11; and vertexes 7, 8, and 9 by vertexes 6, 10, and 11.[14]

Having determined the free vertexes, one can then apply the MSDA principle to the reduced search space. For the case of the simple descent algorithm, the only change to the algorithm is that only the free vertexes are directly modified during the optimization, and that the depth of all of the dependent vertexes are recomputed whenever a free vertex is modified. Applying this *method of free vertexes* to the hexagonal prism reduces the number of iterations from 406 to 39, and the run time from 1306 seconds to 47 (the run time is reduced by a greater proportion than the number of iterations because the DP term has effectively been removed from the objective function).

Thus, the advantage of using the method of free vertexes is that it reduces the search space and run times considerably—oftentimes an order of magnitude or more. The disadvantage of using this approach is that, unlike the planarity enforcing MSDA algorithm, it requires a virtually perfect line drawing of a planar-faced object to ensure that the resulting reconstruction is planar. For example, adjusting the (x, y) coordinates of even one vertex by a small amount in a line drawing such as the cube (example A), can cause the 3D wire frame to be highly nonplanar for some choices of z coordinates of the free vertexes. Consequently, the method of free vertexes can produce reconstructions that are not psychologically plausible. Nonetheless, there are certain situations in which this approach can be effective, both for special kinds of line drawings, and for line drawings that are first processed to make them precise projections of the intended 3D object.

4 Implications for Human Vision

Line drawings provide an effective means of communication about the geometry of 3D objects. It is a matter of some debate as to whether the interpretation of line drawings is a learned skill, or whether line drawings are isomorphic to some intermediate construction of the human visual system (HVS) in its normal processing

of imagery, but in either case an understanding of how humans interpret line drawings is extremely important in enabling man–machine communication with respect to images, diagrams, and spatial constructs. In this section we address two related questions arising out of the investigation described in earlier sections: (a) under what conditions is a line drawing actually given some intended 3D interpretation, and (b) under what conditions does a moving rigid (wire frame) object actually appear rigid.

Some, but not all, line drawings are perceived by human subjects as being three dimensional. What attributes of the drawing promote such an interpretation, and what are the constraints on the nature of the resulting 3D construction? Partially because human introspection is involved, this is a very difficult question to answer. For example, if the drawing is recognized as a known or previously encountered 3D object, it might be visualized this way even though it violates conditions necessary for an unfamiliar object to be perceived as being three dimensional. Gestalt psychologists have suggested that if the drawing offers a simpler construct when seen as three dimensional than when seen as being flat, it will be perceived as being three dimensional; however, an effective computational procedure to evaluate "simpler" has yet to be provided (and there is also the problem of producing the corresponding 3D construct). One might consider that minimizing angular variance is an example of a simplicity principle, but we have not yet been able to define a formal complexity metric, as was done, for example, in the work of Leclerc (1989).

It appears to be much more productive to show a human subject a candidate 3D reconstruction and ask if it corresponds to some given line drawing than it is to tabulate introspective judgments about whether objects appear to be 2D or 3D. The former approach, in fact, is how Marill presents his results to the reader. Obviously, he can not show an actual 3D reconstruction, but only a projection. If he showed the reconstructed object projected without some spatial relocation, then all we have is the original line drawing back again—and no determination can be made; Marill shows two projections of his reconstructed objects, rotated by a few degrees, for evaluation by the reader. Now we know that every orthographic extension is a *geometrically* feasible reconstruction, so on what basis does the human judge acceptability (i.e., what we have called a *psychologically* plausible reconstruction). It is easy to hypothesize a whole list of conditions that should be met—mostly different instantiations of the idea that regularities (such as parallel lines or equal angles and lengths) observed in the line drawing are not accidental, and should be preserved in the reconstructed object; orthographic projective invariants, such as parallelism, should then also be preserved in the reprojections of the spatially relocated object. One could write computational procedures to search for such invariants, but this approach seems incompatible with the universality of the human evaluation process (e.g., none of the invariants we happened to think of may be present in the instances we are considering). A more powerful idea is to require that the computational procedure that produced the original reconstruction give the same result when applied to any of its general position reprojections—that is, a consistency criterion. This is exactly the condition that obtains when we observe a moving or rotating object to be rigid; when we see a (continuous) sequence of projections that we perceive as being isomorphic to the same geometric reconstruction, we perceive the object as being rigid.[15]

Applying the above ideas to an evaluation of the MSDA algorithm, we find two serious deficiencies in

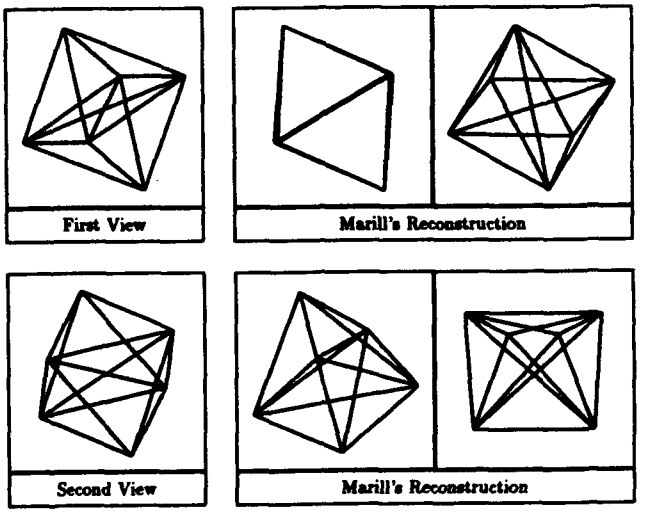


Fig. 10. Illustration of the failure of Marill's algorithm to recover geometrically similar 3D models from two different projections of the same 3D project. The top row shows the input line drawing of the 3D object as seen from one viewpoint (similar to example G), and two views of Marill's reconstructed object. The bottom row shows the input line drawing of the same 3D object as seen from a different viewpoint, and two views of Marill's reconstructed object. The two reconstructed objects not only appear different, but are in fact significantly different geometrically, as we verified by examining their internal representation. In contrast, applying our algorithm to both of these input line drawings, as well as ten other randomly chosen views produced reconstructions with an angular error of less than thirteen degrees from the original object.

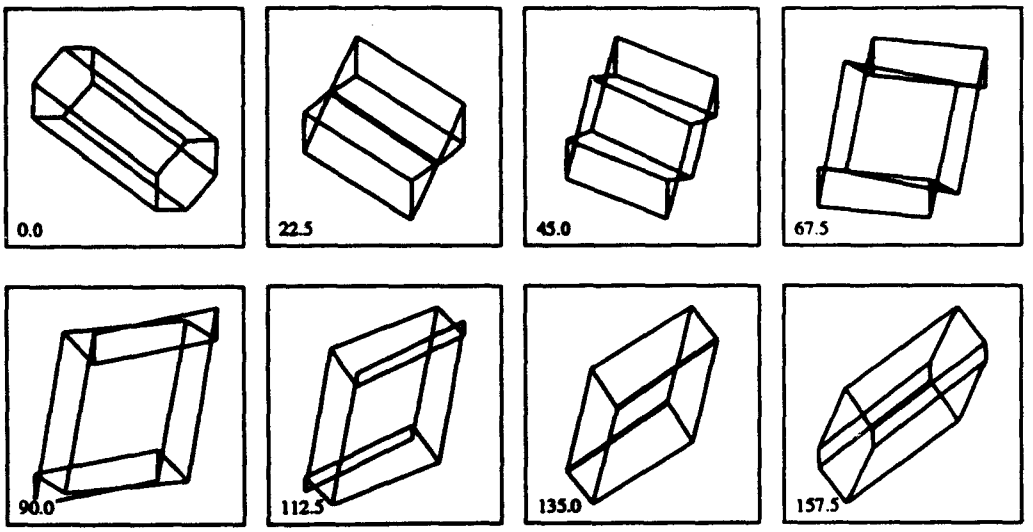


Fig. 11. The illusion of nonrigidity for a rotting wire frame with nonplanar faces. The wire frame, Marill's reconstruction of example J, is rotated about a vertical axis in the center of the object. The rotation angle is written in the lower left-hand corner of each box.

the algorithm. First, when presented with two different orthographic projections of an object, the MSDA algorithm sometimes fails to recover 3D wire frames that are even remotely similar to each other (see figure 10). Second, when we use the computer to create a rotating display of some of the reconstructions obtained with the use of the MSDA algorithm, we see what appears to be the movement of a nonrigid object (see Figure 11).

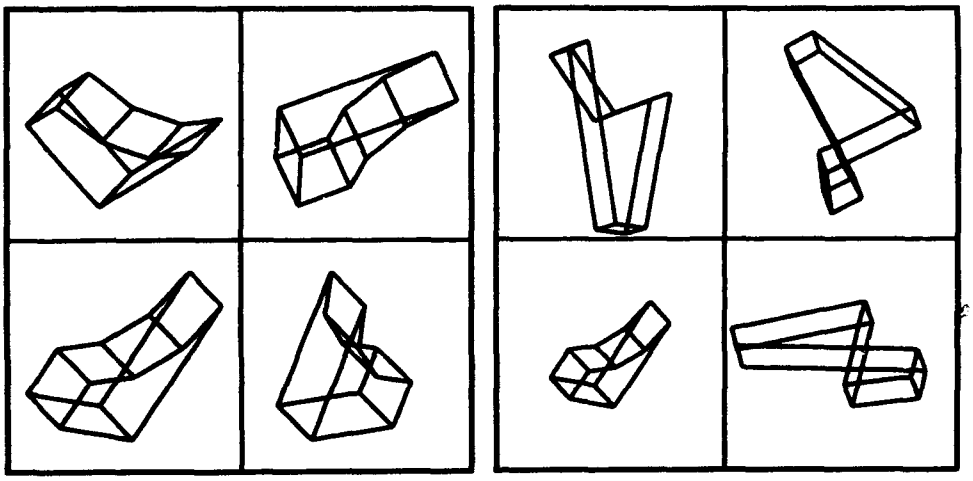
The latter observation led to a number of casual experiments to determine the factors affecting the perception of nonrigidity in displays of rotating 3D wire frames. We found that wire frames with pronounced nonplanar faces (where one would have expected a planar face from the line drawing) appear to be nonrigid. Marill's solution for example I (asymmetric solid) does appear rigid under rotation, even though the faces are slightly warped. However, his solution is very nearly planar; if we force a bit more distortion into the solution, the object then appears to deform under rotation. Thus, it would appear that strict (or at least near) planarity for the appropriate faces is a *necessary* condition for the perception of rigidity.

However, planarity by itself was not sufficient to create a perception of rigidity. For example, if one chooses random values for the free vertexes of a certain line drawing (see section 3.6), one produces an object whose faces are strictly planar. However, unless the resulting figure is also a local minimum of the SDA, the resulting 3D wire frame does not appear rigid when rotated. Similarly, the wire frames of some line drawings with all of the z coordinates set to zero appeared nonrigid when rotated (e.g., example A). Furthermore, all of the hundreds of solutions produced by the planarity enforcing MSDA algorithm that we looked at appeared rigid under rotation. Thus, we tentatively conclude that a wire frame must not only be planar to be perceived as rigid, but must satisfy additional constraints, such as being a local minimum of the SDA.

5 Future Work

There are a number of directions that we have begun to explore or that we plan on exploring in the near future.

The first of these, for which we have some preliminary results, is a redefinition of the objective function in which the angles are partitioned into groups that should be equiangular in 3D. This becomes necessary either when there are angles in the line drawing that are not a part of any planar face or when the angles in a planar face are not all equal in 3D (in either of these cases, the symmetric preference theorem of appendix D does not hold). An example of the first case is the hinge (figure 6), in which angles (1 0 4) and (2 3 5) are not a part of any planar face. An example of the second case is the truncated box (figure 5), in which angles (1 2 3) and (2 3 4) should be equal to each other but not equal to the other angles in planar face (0 1 2 3 4), and similarly for face (5 6 7 8 9).



(a) (b)

Fig. 12. An illustration of the need to group angles together than should be equiangular, rather than applying the planarity enforcing MSDA principle to all angles. (a) The reconstruction using an appropriate equiangular grouping. (b) The reconstruction using the planarity enforcing MSDA principle applied to all angles.

By changing the definition of the SDA term to be the sum of the standard deviation of the angles in each equiangular group (weighted by the number of angles in that group), we have improved the reconstruction of these two objects considerably. Defining a simple, yet robust, set of rules that can automatically determine the equiangular groups for a line drawing, as we did for the planar faces of the line drawings in this paper, is still an open question. A simple rule is to group together all angles that are a part of a convex face. This is illustrated in figure 12. The reconstruction is accurate to 3 degrees, whereas using the SDA over all angles gives a relatively poor reconstruction.

A second direction that we plan on exploring is to implement a preprocessing step that would take a rough sketch and enforce various constraints in 2D, such as (1) parallelism between designated pairs of lines, or between designated lines and axes; (2) equality in length between designated lines, or between lines and fixed lengths; and so forth. The paradigm would be similar to the one for the interpretation of the line drawing, namely some set of rules would be used to determine which lines should be parallel or of equal length (with outside intervention always possible), and an optimization step would then enforce the constraints while moving as little as possible from the original line drawing. The ideal is to be able to do as much of this as possible without intervention for an interactive user.

A third direction is to explore the relationship between what we have done and previous work in understanding the 3D shape of curves, such as (Barrow & Tenenbaum 1981; Stevens 1981; Witkin 1981; Barnard & Pentland 1983; Malik & Maydan 1989; Pentland & Kuo 1990).

An intriguing relationship between Barrow and Tenenbaum's work on single curves and our work on planar faces is as follows. They defined the problem of interpreting curves in a manner similar to the way that we and Marill did: by defining an objective function over the z coordinates of the object and minimizing that objective function using a descent algorithm.[16] Their objective function was the integral of the change in curvature squared plus the torsion squared. Thus, an ideal curve for their objective function is a planar circle, since both terms in the integral are then zero everywhere (when the end-points are removed from the integral, the arc of a planar circle is also an ideal curve for their objective function). Analogously, one of the ideal curves for our definition is a regular planar polygon (or an arc of a regular planar polygon), since then both the SDA and DP are zero. Thus, the similarities are that the SDA plays a role similar to the integral of squared change in curvature, and the DP plays a role similar to the integral of squared torsion. Some of the differences are that both the SDA and DP1 terms are global measures of symmetry and planarity, while the curvilinear measures are integrals of local measures. A second difference is that the SDA is also zero for some nonregular and even nonconvex polygons.

Pentland and Kuo (199) applied Barrow and Tenenbaum's idea to distinctly nonplanar curves and surfaces by leaving out the torsion component. It is somewhat surprising that this worked since both Barrow and Tenenbaum's and our own experience indicate that

planarity is a key ingredient in making the optimization approach work. We will explore this question in the near future.

Finally, we would like to find some computationally effective procedure for using the rigidity under rotation criterion in the 3D recovery process, rather than as a final check on proposed solutions.

6 Discussion

Traditional blocks-world problems are mathematical in nature, they deal with issues of existence and consistency based strictly on geometric considerations; they make no reference to what people actually see. The problem defined by Marill is psychological; since every line drawing has an infinite number of mathematically valid orthographic extensions and no invalid ones, on what basis does the HVS select a particular extension as being psychologically acceptable? Marill proposed an intriguingly simple criterion for duplicating human preference, but we have shown that, while it often produces an acceptable answer, it is unreliable even in very simple situations.

Marill's work has similarities to the Huffman-Clowes-Waltz approach that focused on how polyhedral vertexes can appear in a line drawing and, hence, the constraints such vertexes impose on the implied 3D model; Marill considers only the constraints implied by line intersections at specified vertexes in the line drawing. Mackworth, Kanade, and Sugihara found it necessary to introduce constraints based on the explicit assignment of vertexes to planar faces. We show here the need for introducing a similar explicit requirement for planarity (actually, in the context of optimizing an objective function, our constraint is soft in that it can be violated). However, in our case, the requirement for planarity is justified on psychological grounds rather than as a means for achieving a geometrically more competent algorithm.

The preference of the HVS to interpret a line drawing as the most symmetric polyhedral (planar-faced) object consistent with the drawing is well established in the psychological literature. Marill appeared to have discovered a simple computational procedure for finding such solutions for any given line drawing, but on closer examination, it became apparent that his MSDA principle does not enforce (or even prefer) planar solutions.[17] Because of this deficiency, MSDA is unreliable even in very simple situations. We were able to prove (appendix D) that if a planarity preference is explicitly added to the MSDA objective function, then indeed, the nonobvious preference for symmetric solutions is also present. However, we are now forced to address the problem of how to provide the auxiliary information necessary to partition the drawing into the coherent components corresponding to the 3D planar faces. It appears that the HVS selects some subset of the contours in the line drawing as corresponding to the planar faces in the 3D model, and if we do not supply this information to a recovery algorithm (either explicitly or by providing a set of conditions implying the same information), we will fail to recover psychologically acceptable models.

Most of the work in the blocks-world tradition employed perfect *labeled* line drawings with the assignment of vertexes to faces given as part of the input specifications. If we follow the same approach (although we are not concerned with having perfect line drawings since our recovery method employs optimization, which can tolerate deviations from any of the constraints embodied in the objective function), then we at least have provided a tool for simplifying man-machine communication using the language of line drawings. However, there is obvious theoretical value in understanding the criterion for human selection of the circuits in the line drawing that correspond to planar faces in the 3D model.[18] In part, this importance is related to the issue of how the HVS recovers the shape of a moving object. Even though there are a few well-known exceptions, it is widely believed that the HVS will assume an object to be rigid and correctly recover its shape if this is indeed the case.[19] However, the rigid wire frames with nonplanar faces provide a whole class of counter-examples to this belief—they appear to be nonrigid when observed in motion (even at very low speeds where maintaining correspondence of vertexes from one projection to the next is no problem). The nonrigidity appears to result from the HVS making incorrect decisions about how the drawing can be partitioned into planar faces (see appendix E).

7 Summary

Marill's recently published paper claimed that the simple procedure he described could duplicate human judgment in recovering the 3D wire frame geometry of objects depicted in line drawings. He provided some impressive examples, but no theoretical justification to

back his claims. In this article, we critically examined the merits of Marill's algorithm, provided at least a partial explanation for its competence, identified weaknesses, showed how it could be improved, and discussed the implications of this work for clarifying some important problems in human perception.

In particular, we provided a number of theorems that show that minimizing the standard deviation of angles is (potentially) a simple and effective method for selecting symmetric solutions when the constraining line drawing (which is the projection of a wire frame that may be incomplete) permits such interpretation. On the other hand, we showed that Marill's algorithm could fail in simple cases, that he employed an optimization procedure that was often too weak to find the correct answer even when it was within the competence of the objective function, and that the algorithm would often produce wire frames with nonplanar faces (something no human would intuitively accept in perceiving a straight-line drawing as a 3D configuration).

We argued that an important condition in testing or evaluating the psychological plausibility of a reconstruction is that its reprojections (after spatial relocation) result in the same object being produced by the recovery algorithm. For the human visual system, this is equivalent to the condition that the recovered object appear rigid when observed during movement or rotation. The perception of rigidity for wire frames appears to be highly correlated with the presence or absence of strongly nonplanar faces. By modifying Marill's objective function to explicitly favor planar-faced solutions, and by using a more competent optimization technique, we were able to demonstrate significantly improved performance in all of the examples Marill provided as well as those additional ones we constructed ourselves. The robustness of our algorithm was demonstrated by obtaining consistent psychologically plausible reconstructions in hundreds of experiments involving variations in viewpoint and initial conditions for the approximately 20 objects in our database.

Acknowledgements

The work reported here was partially supported by the Defense Advanced Research Projects Agency. We gratefully acknowledge the valuable discussions with Aaron Bobick and Thomas Strat regarding both the content and organization of this article. A number of improvements and clarifications in its final version were suggested by Thomas Marill in a private communication.

Notes

1. Gradient space, originally conceived of by James Clerk Maxwell in 1864 (see (Whitely 1986)) and rediscovered by D.A. Huffman, provides only necessary conditions for planar realizability of general polyhedral objects with hidden lines removed, and thus consistent edge labeling is possible for impossible blocks world and Origami objects. Further, the labeling/recovery algorithms were not always competent to find an existing solution.
2. There were some other problems of lesser significance for our purposes. For example, the algebraic formulation was sensitive to computation round-off errors, and digitization errors in specifying the line drawing; a realizable object could be rejected because of such minor numeric inaccuracies. Sugihara dealt with this problem by adding an optimization step to his algorithm, which could find a feasible reconstruction if the input drawing was an almost correct specification.
3. Marill, on the other hand, calls the set of all possible zs the orthographic extension.
4. For simplicity, the vertexes are represented by only two digits of precision in the table. However, we used the full 32-bit precision of the projection in the internal representation used by the algorithms.
5. We note that while there generally can be many different ways of covering a line drawing, those of blocks-world objects with hidden lines removed will be covered uniquely if we demand that the interior of the 2D circuits be free of any lines. We also note that it is not always possible to cover a line drawing with *simple closed* circuits corresponding to the specified planar faces of a given orthographic extension (see example N). It may also be the case that a given covering has no nontrivial orthographic extension with planar faces as specified, as in example O.
6. One face, in example H, is an exception to this statement. However, there are enough other geometric constraints in this particular case to enforce planarity.
7. Because this rule typically produces many additional planar faces, it was not used in figures 2 through 6. For these line drawings, the results are virtually identical with or without these additional planar faces. However, the rule was used in the stability and robustness experiments of section 3.4.
8. The SDA term is first squared to make it commensurate with the DP term. Note that squaring the SDA term has no effect on the minimization when $\lambda = 1$ (i.e., the simple MSDA algorithm), because the SDA term is positive, and squaring is a monotonic function of the positive reals.
9. This assumes the line drawing is perfect. We later discuss how such perfect drawings can be obtained in an interactive environment.
10. Since we had only the original line drawing for each of Marill's examples, we used the reconstruction from each line drawing as the 3D object for the random projections.
11. The line lengths for these drawings were approximately in the range of 2 to 5.
12. For all line drawings except the truncated box and the hinge, the largest absolute difference in angles between any trial and the reconstruction with $z = 0$ was less than one degree. For the truncated box, the largest error was less than fifteen degrees. For the hinge, one of the trials caused the hinge to "fold" with arc-pairs (1 0 4) and (2 3 5) going to zero degrees. Otherwise, the largest error was less than seven degrees.
13. Modulo a change in sign in the z coordinates.

14. The set of free vertexes is by no means unique. For example, any three vertexes from one hexagon plus any vertex from the other hexagon will do for this line drawing. We have a simple algorithm for finding a set of free vertexes, but have not yet proven that it is correct, so we do not present it here.
15. The successive reconstructions are not independent; to the extent that they allow a range of interpretations, the parameters selected for one interpretation will influence the parameter selections for successive interpretations.
16. Some differences are that Barrow and Tenenbaum considered arbitrary, but *known* perspective transforms in their paper, while Marill used only orthographic projections. In either case, the set of state variables is equivalent. In addition, Barrow and Tenenbaum did not consider the use of a continuation method.
17. Marill, of course, only returns the wire frame. But in the case of a blocks-world object, competent algorithms exist for finding all the valid completions of the wire frame as a solid polyhedral object (Strat 1984, Markowsky and Wesley 1981).
18. As noted in section 3.1 and appendix A, we have made some initial progress toward the solution of this problem and have developed an algorithmic procedure that can successfully handle all of the examples discussed in this paper, but we recognize that this is still far short of a complete solution.
19. For example, by using Ullman's result that three distinct orthographic projections of four noncoplanar points in a rigid configuration are sufficient to uniquely determine the structure and motion up to a reflection about the image plane.

References

Barnard, S.T., and Pentland, A.P., 1983. Three-dimensional shape from line drawings, *Proc. 8th Intern. Joint Conf. Artif. Intell.*, Karlsruhe, W. Germany.

Barrow, H.G., and Tenenbaum, J.M., 1981. Interpreting line drawings as three-dimensional surfaces, *Artificial Intelligence* 17(1-3):75-116.

Clowes, M.B., 1971. On seeing things, *Artificial Intelligence* 2(1):79-116.

Draper, S.W., 1981. The use of gradient and dual space in line-drawing interpretation, *Artificial Intelligence* 17:461-508.

Hochberg, J., and McAlister, E., 1953. A quantitative approach to figure "goodness." *J. Exp. Psychol.* 46:361-364.

Huffman, D.A. 1971. Impossible objects as nonsense sentences. In Meltzer and Michie, ed., *Machine Intelligence* Vol. 6, p. 295-323, Edinburgh University Press.

Kanade, T., 1980. A theory of Origami world, *Artificial Intelligence* 13(1):279-311.

Leclerc, Y.G. 1989. Constructing simple stable descriptions for image partitioning. *Intern. J. Comput. Vis.* 3(1):73-102.

Mackworth, A.K., 1973. Interpreting pictures of polyhedral scenes, *Artificial Intelligence* 4(2):121-137.

Malik, J., and Maydan, D., 1989. Recovering three-dimensional shape from a single image of curved objects, *IEEE Trans. Patt. Anal. Mach. Intell.* 11(6):555-566.

Marill, T., 1991. Emulating the human interpretation of line-drawings as three-dimensional objects, *Interna. J. Comput. Vis.* 6(2):147-161.

Markowsky, M.A., and Wesley, G., 1981. Fleshing out projections, *IBM J. Res. Develop.* 25(6):934-954.

Pentland, A., and Kuo, J., 1990. Three-dimensional line interpretation via local processing, *Electronic Imaging* 1249-30. Also Media Lab Technical Report 131.

Pomerantz, J.R., and Kubovy, M., 1981. Perceptual organization: an overview. In *Perceptual Organization* pp. 423-456, Lawrence Erlbaum Associates; Hilladale, NJ.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., 1986. *Numerical recipes, the art of scientific computing*, Cambridge University Press, Cambridge.

Stevens, K.A., 1981. The visual interpretation of surface contours, *Artificial Intelligence* 171(1-3):47-74.

Strat, T.M., 1984. Spatial reasoning from line drawings of polyhedra, *Proc. DARPA Image Understanding Workshop*, New Orleans, p. 230-235.

Sugihara, K., 1982. Mathematical structures of line drawings of polyhedrons—toward man-machine communication by means of line drawings, *IEEE Trans. Patt. Anal. Mach. Intell.*, 4(5):458-469.

Sugihara, K., 1984. A necessary and sufficient condition for a picture to represent a polyhedral scene. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(5):578-586.

Waltz, D.A., 1972. Generating semantic descriptions from line drawings of scenes with shadows. Technical Report AI-TR-271, MIT.

Whitely, W., 1986. Two algorithms for polyhedral pictures, *Proc. 2nd Annual Symp. Computa. Geometry*, p. 142-149.

Witkin, A.W., 1981. Recovering surface shape and orientation from texture, *Artificial Intelligence* 17:17-47.

Witkin, A.W., Terzopoulos, D., and Kass, M., 1987. Signal matching through scale space, *Intern. J.. Comput. Vis.* 1:133-144.

Appendix A. Psychological Assumptions

The following are some of the basic assumptions that we believe are typically made by people in the reconstructions of wire frames from line drawings, and some constraints relevant to partitioning a line drawing into planar face. They are known to have rare exceptions.

1. Three dimensional wire frames, derived from line drawings, have implied planar faces inside subsets of their closed circuits; they *can* also have struts, such as legs or bracing wires, in or on a planar face. (Strongly nonplanar faces produce psychologically implausible solutions.)
2. Symmetric reconstructions are preferred over nonsymmetric ones.
3. Parallel lines in a line drawing are parallel in space. Lines connecting vertexes falling on two parallel lines are in a common plane with the two parallel lines.
4. Many-sided convex closed contours without *internal circuits* (in a 2D line drawing) are likely to correspond to the contours of planar faces in the corresponding 3D orthographic extension (see B4). An

internal circuit to a convex polygon is defined to be a circuit for which all the vertexes are internal to the polygon, and for which the ends of the circuit lie on nonadjacent vertexes of the polygon.

5. A closed simple contour in a line drawing, without internal lines, corresponds to a planar face in the corresponding 3D reconstruction.

An algorithmic procedure for identifying 3D planar faces in the corresponding 2D line drawing of a wire frame has been constructed by composing the requirements of items 3, 4, and 5 into a single algorithm, as defined in section 3. That procedure is sufficient to deal with all of the examples we discuss here, but is not general enough to handle other cases we can think of.

Appendix B. Projective Invariants

The following are some important projective invariants for planar geometric structures.

1. The sum of the interior angles (measured between 0 and 360 degrees) of a closed planar contour with n sides equals $(n - 2)$ 180 degrees. Thus, since a polygon of n sides projects to a polygon of n sides under both orthographic and central projection, the mean value of the interior angles of a given closed planar contour $[(n - 2)180/n]$ is invariant under both orthographic and central projection.

We note that Marill measures angles only in the interval between 0 and 180 degrees. To the extent that we are primarily concerned with equiangular closed contours in the application of the above theorem in explaining and using his results, this discrepancy is irrelevant since all the interior angles of such contours are less than 180 degrees.

2. Consider an angle (two line segments sharing a common endpoint) in 3D space and its orthographic projection. We will call the plane containing the angle the source plane, and the plane containing its projection the projection plane. If the angle is translated in the source plane, its projection is also translated, but does not change in magnitude from its original projected value. Now consider a set of n angles lying on a common source plane, such that the sum of these angles is 360 degrees. If it is also the case that the angles can be translated so that when all their vertexes coincide, they exactly span an angle of 360 degrees, then the mean value of the set of angles $(360/n)$ is unaltered under orthographic projections. We will call such a collection of angles a "complete-star." (Example C, for instance, contains a complete-star consisting of the eight 45-degree angles formed at the corner vertexes by the diagonals with the sides of the square. Example G contains this same configuration in its central plane.) We note that if an essentially infinite number of copies of an angle of d degrees (where $360/d = k$ and k is an integer) is uniformly distributed in orientation over a plane, then the mean value of the angles under any orthographic projection of the plane is the constant value d.
3. We note that if the angle between two line segments is less than 180 degrees, the angle can be closed to form a triangle, and since triangles are preserved under both orthographic and central projection, an angle of less than 180 degrees will never transform under such projections into one of more than 180 degrees. We will call a closed planar contour convex if the region it bounds is convex. Since a convex contour has all internal angles of less than 180 degrees, a convex planar contour remains convex under both orthographic and central projection.
4. We note that the orthographic projection of an arbitrary nonplanar polygonal space curve, with four or more sides, has a probability of projecting to either a nonsimple or concave curve with a probability (P) that increases with the number of sides:

$$P > 1 - 0.5^{n-3} \qquad \text{for } n \geq 4$$

This expression is based on the following model: Consider a process that generates a chain of 3D random vectors by generating three random numbers for each vector (in spherical coordinates, an angle uniformly distributed between 0 and 360 degrees, a second angle between 0 and 180 degrees, and a length uniformly distributed between 0 and some fixed integer L). As each vector is generated we extend the projection of the developing space curve on the X-Y image plane. The process stops after some fixed number of steps, which is determined by choosing a random number in some given range; the curve is now closed by connecting the starting point, which could be the origin of the X-Y plane, to the last point generated and this determines whether the inside is to the left or right as we follow the chain of edges of the projected polygon. We note that the only relevant factor in whether the projected closed contour is convex or concave is the cylindrical angle giving the rotation of each of the random vectors relative to the X axis in the image plane. For more than three sides, there is a 50% probability at each vertex that the inside angle is greater than 180 degrees,

which thus produces a concave polygon (the last closing side can be ignored since it does not have the same statistics as the other edges in our random model.) Other probabilistic models would give nonidentical, but similar results. The > condition is based on additional considerations, such as the projected curve intersecting itself even though the input specification does not record a vertex at the cross-point.

5. Closed four-sided polygonal space curves with 90-degree angles at each vertex are planar contours. To prove this assertion, let the sequence of vertexes be labeled a, b, c, and d. Let the plane containing lines L_{ab} and L_{bc} (and thus vertexes a, b, and c) be called P_1. Since all angles are 90 degrees, L_{cd} must lie in a plane (P_2) normal to L_{bc} at c. Similarly, L_{ad} must lie in a plane (P_3) normal to L_{ab} at a. Vertex d must then lie on the line (L_d of intersection of P_2 and P_3, which is normal to P_1. We know one solution is to locate d at the point of intersection (d^*) of L_d and P_1 (where a, b, c, and d^* form a rectangle). This is the planar solution and we wish to show that no other solution is possible. We note that a second constraint on the location of d is that it must lie on a sphere with diameter ac (i.e., all right angles, with legs passing through points a and c, must be inscribed angles of circles through a and c with diameter ac). We know d^* lies on the sphere and P_1 is a bisecting plane of the sphere. Thus L_d is tangent to the sphere at d^* and d^* is the only possible solution.
6. *A Global Planarity Test for a Space Curve.* A planar polygonal curve has a sum of internal angles equal to $(n - 2)180$ degrees. Thus, if the curve is triangulated using only the existing vertexes along the curve, the sum of the angles of the triangles is also $(n - 2)180$.

Case 1: Consider a space curve S that projects to a convex planar curve. If the space curve is itself planar, the sum of its angles (measured between 0 and 180 degrees) is $(n - 2)180$. Assume S is nonplanar, that is, there is a "fold" along one or more edges of *some* triangulation of its planar projection. Consider the vertex V at the intersection of one such fold (with respect to the implied triangulation T) and S. The plane through the two edges of S meeting at V, and the faces of the triangles of T that have edges intersecting at V, form a polyhedral angle. It is known that any face angle of a polyhedral angle is less than the sum of the other face angles. Therefore, the sum of the angles of the space curve is equal (at vertexes with no folding) or less (at vertexes with folding) than the sum of the angles of the triangles in T (i.e., less than $(n - 2)180$).

Case 2: If the projection of the space curve S is concave, and we measure angles between 0 and 360 degrees, the sum of the *internal* contour angles in the planar projection will equal $(n - 2)180$ as in Case 1. However, while the space angles with projections of less than 180 degrees will decrease at folds, the internal angles greater than 180 degrees will increase (i.e., at vertexes where there are folds, the polyhedral angle in the argument given in Case 1 is now formed for the *external* angle of S at V). Thus, since some angles will increase and others decrease, we cannot be sure that the curve is planar even if the sum of its internal angles equals $(n - 2)180$. However, we do have a sufficient condition for nonplanarity. That is, the curve is known to be nonplanar if the sum of its internal angles, measured between 0 and 360 degrees, is not equal to $(n - 2)180$ degrees.

Appendix C. A Partition Theorem

The variance of a set of S of n objects $\{a_i\}$ is defined as

$$V = \frac{1}{n}\sum_{i=1}^{n}(a_i - M)^2 = \left[\sum_{i=1}^{n}\frac{a_i^2}{n}\right] - M^2$$

where

$$M = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Let us now partition the $\{a_i\}$ into k subsets, such that the subset S_j has n_j elements and mean M_j where:

$$M_j = \frac{1}{n_j}\sum_{S_j} a_i$$

Let V_j be the variance of S_j about M_j and let $\Delta_j = (M - M_j)$.

Theorem:

$$V = \frac{1}{n}\sum_{j=1}^{k} n_j[V_j + \Delta_j^2]$$

Proof: The expression for V can be rewritten as

$$V = \frac{1}{n}\left[\sum_{S_1}[a_i - (M_1 + \Delta_1)]^2 + \sum_{S_2}[a_i - (M_2 + \Delta_2)]^2 + \cdots + \sum_{S_k}[a_i - (M_k + \Delta_k)]^2\right]$$

If we let:

$$V_j^* = \sum_{a_j} [a_i - (M_j + \Delta_j)]^2$$

Then we have:

$$\frac{V_j^*}{n_j} = \sum \frac{a_i^2}{n_j} + M_j^2 + \Delta_j^2 - 2\Delta_j \sum \frac{a_i}{n_j} - 2M_j \sum \frac{a_i}{n_j} + 2M_j \Delta_j$$

Given that $\Sigma(a_i/n_j) = M_j$, we note that the 4th and 6th terms cancel and the 2nd and 5th terms combine:

$$\frac{V_j^*}{n_j} = \left[\sum \frac{a_i^2}{n_j} - M_j^2\right] + \Delta_j^2 = V_j + \Delta_j^2$$

And

$$V_j^* = n_j[V_j + \Delta_j^2]$$

QED

Appendix D. Symmetric Preference Theorem

Recall that

1. In appendix B we showed that the average angle of all planar orthographic extensions of a given simple closed 2D contour are the same, and that the average angle of all planar orthographic extensions of a complete-star are also the same;
2. in appendix C we proved a theorem that allows us to compute the SDA of a set of simple closed planar contours (and/or complete-stars) as the sum of two components. The first component is the variance of the angles in a contour or star about the mean angle of that contour or star, summed over all contours and stars. The second component is a weighted sum of the squared differences between the mean angle of each contour and star, and the average of all the angles under consideration.

By (1), the second component of the variance is constant over all planar orthographic extensions because (a) the mean of each contour and star is constant over all such extensions, and (b) the mean of all angles can be computed as the weighted sum of the mean of each contour and star.

Consequently, if we restrict our attention to the planar orthographic extensions of a line drawing, then by (2) above, only the first component of the variance will change over the extensions. Since the first component is zero for an extension comprising only equiangular planar contours and stars (such as the solutions for examples A, B, C, G, J, K, and L), and since it is positive otherwise, then such symmetric solutions correspond to the global minimum of the SDA over all planar orthographic extensions.

Appendix E. Factors Affecting the Perception of Nonrigidity

If we rotate a *randomly* derived orthographic extension of almost any of the line drawings used as examples in this article, the object appears nonrigid to most observers (even though, of course, the wire frame is actually a rigid object). While there are many possible explanations for this phenomenon, our conjecture is that it is primarily due to special position projections of the wire frame (that occur at one or more poses in its rotation) that lead the HVS to incorrectly assume that some projective invariant (such as parallel lines, see figure 11) is being observed. This, in turn, causes incorrect expectations about the presence and location of planar faces.

We informally looked at some other possible causative factors, but did not observe consistent nonrigidity phenomena. For example, we looked at objects, such as example N that produce compelling 3D interpretations with Necker reversals, but for which the drawing is incomplete—it does not show all the edges that should be visible, for example, where planar faces intersect. There was the possibility that these missing edges in the 3D model (and thus missing lines in the drawing) could cause the appearance of a nonplanar-faced object to be observed. But the hinge, and the few other objects we looked at in this category, appeared rigid.

We also looked at nonplanar orthographic extensions of drawings that generally appeared flat, including blocks-world type drawings that do not have corresponding polyhedral realizations (such as example O). The results here were ambiguous. The rotating objects generally produced illusions of nonrigidity, but since these objects did not always appear 3D, the illusions were generally very weak.

Some other causal experiments include cases where all the lines connecting the vertexes of the wire frames are deleted; we observed that some of the wire frames that originally appeared nonrigid now appeared to be rigid under rotation. And, as a general observation, we have not encountered any examples in which the wire frame of a (nondegenerate) blocks-world object appears nonrigid when in motion.

Appendix B:

Saliency Detection and Partitioning Planar Curves

Saliency Detection and Partitioning Planar Curves*

Martin A. Fischler and Helen C. Wolf

Artificial Intelligence Center
SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
(fischler@ai.sri.com wolf@ai.sri.com)

Abstract

This paper summarizes the underlying ideas and algorithmic details of a computer program that performs at a human level of competence for a significant subset of the *curve partitioning* task. It extends and "rounds out" the technique and philosophical approach originally presented in a 1986 paper by Fischler and Bolles. In particular, it provides a unified strategy for selecting and dealing with interactions between salient points, even when these points are salient at "different scales of resolution." Experimental results are described involving on the order of 1000 real and synthetically generated images.

Index Terms: computer vision, salient points, critical points, *curve partitioning*, *curve segmentation*, curve description

1. Introduction

A critical problem in machine vision is how to break up (partition) the perceived world into coherent or meaningful parts prior to knowing the identity of these parts. Almost all current machine vision paradigms require some form of partitioning as an early simplification step to avoid having to resolve a combinatorially large number of alternatives in the subsequent analysis process. Given this critical role for partitioning as a functional requirement of a complete vision system, it is a major challenge to find some significant subset of the partitioning problem for which an algorithmic procedure can duplicate normal human performance. This paper (a compressed version of a much longer document which will appear in IEEE PAMI later this year) summarizes the underlying ideas and algorithmic details of a computer program which performs at a human level of competence for a significant subset of the *curve partitioning* task. It extends and "rounds out" the technique and philosophical approach originally presented in a 1986 PAMI paper by Fischler and Bolles [Fischler86]. For example, it provides a unified strategy for resolving conflicts in selecting among neighboring potential partition points that may be salient at different "scales of resolution."

While our focus in this paper is on curve partitioning in a generalized setting (the curves in our experiments are mostly without semantic meaning), and where the criterion for success is duplicating normal human performance, finding salient points on image curves (potential partition points) plays a critical role in both two and three dimensional object recognition, in curve approximation, in tracking moving objects, and in many other tasks in machine vision.

In many approaches to 2-D object recognition, objects are represented by their boundaries, and the recognition techniques depend (directly or indirectly) on locating distinguished points along the boundary; typically these *distinguished points* are *discontinuities* or extrema of local curvature (sometimes called "corner points") and inflection points [e.g., Mokhtarian86]. "Corners" on the contours of imaged objects are often used as features for tracking the motion of these objects and for computing optical flow [e.g. Mehrotra90]. In 3-D recognition, partitioning is typically one of the first analysis steps – especially when objects can occlude each other. Hoffman and Richards [Hoffman82] argue that when 3-D parts are joined to create complex objects, concavities will generally be observed in their silhouettes, and that segmentation of image contours at concavities (the maxima of negative curvature along the contours) is a good strategy to decompose (even unmodeled) objects into their "natural parts."

In cartography, computer graphics, and scene anaysis, it is often desirable to partition an extended boundary or a contour into a sequence of simply represented primitives (e.g., straight line segments or polynomial curves of some higher degree) to simplify subsequent analysis and to minimize storage requirements [e.g., Teh89].

In our own current work concerned with delineating linear structures in aerial images, the technique presented in this paper was an essential component of the system (briefly described in Appendix C) that produced the results displayed in Figure 6.

*This work was performed under contracts supported by the Defense Advanced Research Projects Agency.

2. Problem Statement

In its most general sense, partitioning involves assigning, to every element of a given "object" set, a label from a given "label" set. For our purposes in this paper, the object set is the set of points along a curve (or contour segment) lying in a prescribed region of a two-dimensional plane. While we deal with cases where the points in the object set do not form a continuous digital curve, in most of our exposition in this paper we will assume that the curves are continuous [1] and non-intersecting. Our label set is binary, points will be called either significant (critical) or non-significant, for some specified purpose. In Fischler and Bolles [Fischler86], it is demonstrated (or at least argued) that *perceptual* partitioning is *not* independent of some assumed task or purpose. In this paper we focus on one of the three tasks discussed in the above reference: Selecting a small number of points (called *critpts*) along a curve segment which could be used as the basis for *reconstructing* the curve at some future time. Figure 1 shows the specific instructions and curves used in one set of relevant experiments involving human subjects; this figure also shows the critpts that were selected by the subjects, and the comparable results produced by our algorithm (called the Saliency Selection System, or SSS, and discussed in Appendix B).

In order to separate the generic partitioning criteria used by human subjects from criteria based on their past experience, such as when the subject is able to assign a name to the curve (e.g., the curve looks like the letter "s"), we used "random" curve segments for our experiments; the technique employed to generate the segments is described in Appendix A. We also wanted to avoid having to deal with the recognition of global features (e.g., symmetry or repeated structure, or even straight lines and analytic curves) as a condition for making critpt selections; avoiding this problem is justified if we are correct in our belief that local and global analysis are accomplished by separate mechanisms. In order to deal with global features, the complexity of any solution would be expanded enormously since a whole new vocabulary of such features and their representations would have to be implemented. The generation and use of random curves took care of this problem also (i.e., it is highly unlikely that symmetries or repeated structure would ever be generated by our random process).

3. Relevance, Prior Work, and Critical Issues

The partitioning problem has been a subject of intense investigation since the earliest work began in machine vision. It has been widely assumed that in order to reduce the combinatorics of scene analysis to a manageable level, it is necessary to decompose images into their meaningful component parts as one of the first steps in the analysis process. The difficulty arises from the need to partition the image into parts before we know the identity of those parts. The underlying assumption then is that there are generic criteria, independent of the goal of the analysis, that if discovered, could be used to obtain useful (or at least, intuitively acceptable) partitioning; additional problem dependent criteria could be always added to produce a more relevant result for some particular purpose.

The partitioning problem becomes progressively harder as we increase the number of dimensions in which we are working; in this paper we only address the 1.5-D problem of partitioning planar curves. A specific criterion which can form the basis of such partitioning was originally proposed by Attneave [Attneave54] – points at which the curve bends most sharply are good partition points. [2] This idea has been the starting point for most of the subsequent efforts in curve partitioning, but attempts to convert this abstract concept into a computationally executable procedure, that gives intuitively acceptable results, has meet with limited success. [3] References [Imai86, Mokhtarian86, Pavlidis74, Rosenfeld73, Teh89, Wuescher91] are representative of work in this area. [4]

The main problems we must solve are:

(a) A way of assigning a measure (or degree) of saliency/criticality [5] to each point on a curve. Most investigators have equated sharp bending of a curve with the mathematical concept of curvature, but curvature is not well-defined for a finite sequence of points (which is how our sensor acquired curves are generally represented). Further, it is not obvious that the mathematical definition of curvature is the best computational approximation to the human criteria for criticality. In Fischler and Bolles [Fischler86], bending is interpreted

[1] Each point of the non-branching one pixel wide curve, with coordinates (x,y), has one or more neighbors with x-coordinates in the set (x+1, x, x-1), and y-coordinates in the set (y+1, y, y-1).

[2] Hoffman and Richards [Hoffman82] give convincing evidence that we should distinguish between positive and negative curvature maxima. That is, on closed curves, extreme points of negative curvature – associated with object concavities – have greater utility as partition points than positive curvature maxima, but the positive maxima (and inflection points) play an important role in describing the individual segments.

[3] As noted later, most of the work on the curve partitioning problem, especially recent work, has *not* been concerned with duplicating generic human performance, but rather with performing specific visual tasks having different criteria for success.

[4] The approach taken by Wuescher and Boyer is distinct in that they first extract contour segments of approximately constant curvature and then *infer* the location of partition points as a secondary operation.

[5] We will use the terms *saliency* and *criticality* somewhat interchangeably in this paper. However, saliency can be considered to be the generic subset of points that are critical for some partitioning task.

as deviation from straightness – it is closely related to proposed approximations to mathematical curvature, as illustrated in Figures 2 and 3, but has a number of advantages: it is an easily measured quantity, even for digital curves (i.e., sequences of coordinate pairs), and as discussed in the next section, its local extrema are in better accord with human preference (choices based on approximations to the definition of mathematical curvature occasionally include anomalous points as shown in the examples of Figures 2 and 3).

(b) A way of adjusting the criticality of a given curve-point to take into account its interactions with its neighbors; i.e., local context. It is obvious that human subjects will often avoid assigning a critpt label to both members of a pair of points, even when both points have high (independent) criticality values, if the points are close neighbors along the curve. The basic approach of local non-maximum suppression is not sufficient, in itself, to duplicate human performance.

(c) A way of dealing with the interactions between critpts that are significant at different scales of resolution. If a human subject looks through a fixed sized window at the same curve segment displayed at two different magnifications, the selected critpts will not always be the same, and the selection at the lower resolution will not always be a subset of those at the higher resolution (e.g., Figure 4). This is in contrast to the commonly held assumption that critpt assignment should be independent of "scale of resolution."

(d) A threshold of significance; a minimal level of criticality below which variations are considered to be noise and no critpt designations are made. (Some investigators reject the idea that any user supplied parameters or thresholds should be necessary.)

We have addressed the above issues through the solutions to a set of subproblems:

1. Definition of an algorithmic procedure (which is parameterized to deal with noise and scale) for assigning criticality values to each point on a curve independent of decisions made about the locations of (other) critpts. The solution to this problem, essentially the procedure given in Fischler and Bolles [Fischler86], provides answers at a human level of performance for isolated critpts (i.e., along a section of a random curve, generated as described in Appendix A, for which human subjects select only one critpt). Thus, for the domains we experimented with (and especially the domain defined in Appendix A), we were able to assign fixed values to scale/resolution and noise/significance parameters so that our program would make the same selections as human subjects when there was near unanimous agreement among these subjects. This algorithm is described in Appendix B.

2. An analysis of how geometric scaling of the input curve, and resolution specific operations on the curve, can be equated, and thus the development of a basis for normalizing criticality scores across scale.

3. Development of a general approach to the problem of resolving the competition/cooperation interactions of geometrically related objects based on "local dominance." The same machinery used to deal with interactions at a given scale of resolution is also used to resolve conflicts across different scales of resolution.

In the remainder of this paper, we describe our solutions to the problems enumerated above, and then present examples and experimental results to justify the design decisions we made and to illustrate the performance capabilities of our algorithm.

4. Evaluation of Saliency

Saliency is a critical attribute (for description and recognition) assigned to perceived things in the world by the human visual system (HVS). While an elusive concept in general, task specific specializations of this concept are easily found that elicit consistent choices across human subjects. An acceptable computational definition of contour/curve saliency must provide [6]

- The specification of a procedure that quantifies the abruptness and extent of the deviation of a curve from its straight-line continuation; a sharp bend is more salient than a shallow one, and the greater the excursion, the more prominent/salient the "feature."

- Agreement with human judgement in terms of both selection, and accuracy of placement, of the critical points (in some well defined context).

4.1 A Computational Definition of Saliency

Conventional definitions of curvature present a number of serious problems with respect to their use as a saliency measure in computational vision (CV). First, the mathematical definition is based on the properties of a curve in the infinitesimal neighborhood about the

[6] In this paper we are primarily concerned with saliency based on *local* cues; locations on a curve where there is a transition from one type of curvature behavior to another, e.g. from perfectly straight to "wiggley," may also be psychologically salient, but such forms of *global saliency* are beyond the scope of our current investigation.

as deviation from straightness – it is closely related to proposed approximations to mathematical curvature, as illustrated in Figures 2 and 3, but has a number of advantages: it is an easily measured quantity, even for digital curves (i.e., sequences of coordinate pairs), and as discussed in the next section, its local extrema are in better accord with human preference (choices based on approximations to the definition of mathematical curvature occasionally include anomalous points as shown in the examples of Figures 2 and 3).

(b) A way of adjusting the criticality of a given curve-point to take into account its interactions with its neighbors; i.e., local context. It is obvious that human subjects will often avoid assigning a critpt label to both members of a pair of points, even when both points have high (independent) criticality values, if the points are close neighbors along the curve. The basic approach of local non-maximum suppression is not sufficient, in itself, to duplicate human performance.

(c) A way of dealing with the interactions between critpts that are significant at different scales of resolution. If a human subject looks through a fixed sized window at the same curve segment displayed at two different magnifications, the selected critpts will not always be the same, and the selection at the lower resolution will not always be a subset of those at the higher resolution (e.g., Figure 4). This is in contrast to the commonly held assumption that critpt assignment should be independent of "scale of resolution."

(d) A threshold of significance; a minimal level of criticality below which variations are considered to be noise and no critpt designations are made. (Some investigators reject the idea that any user supplied parameters or thresholds should be necessary.)

We have addressed the above issues through the solutions to a set of subproblems:

1. Definition of an algorithmic procedure (which is parameterized to deal with noise and scale) for assigning criticality values to each point on a curve independent of decisions made about the locations of (other) critpts. The solution to this problem, essentially the procedure given in Fischler and Bolles [Fischler86], provides answers at a human level of performance for isolated critpts (i.e., along a section of a random curve, generated as described in Appendix A, for which human subjects select only one critpt). Thus, for the domains we experimented with (and especially the domain defined in Appendix A), we were able to assign fixed values to scale/resolution and noise/significance parameters so that our program would make the same selections as human subjects when there was near unanimous agreement among these subjects. This algorithm is described in Appendix B.

2. An analysis of how geometric scaling of the input curve, and resolution specific operations on the curve, can be equated, and thus the development of a basis for normalizing criticality scores across scale.

3. Development of a general approach to the problem of resolving the competition/cooperation interactions of geometrically related objects based on "local dominance." The same machinery used to deal with interactions at a given scale of resolution is also used to resolve conflicts across different scales of resolution.

In the remainder of this paper, we describe our solutions to the problems enumerated above, and then present examples and experimental results to justify the design decisions we made and to illustrate the performance capabilities of our algorithm.

4. Evaluation of Saliency

Saliency is a critical attribute (for description and recognition) assigned to perceived things in the world by the human visual system (HVS). While an elusive concept in general, task specific specializations of this concept are easily found that elicit consistent choices across human subjects. An acceptable *computational* definition of contour/curve saliency must provide [6]

- The specification of a procedure that quantifies the abruptness and extent of the deviation of a curve from its straight-line continuation; a sharp bend is more salient than a shallow one, and the greater the excursion, the more prominent/salient the "feature."

- Agreement with human judgement in terms of both selection, and accuracy of placement, of the critical points (in some well defined context).

4.1 A Computational Definition of Saliency

Conventional definitions of curvature present a number of serious problems with respect to their use as a saliency measure in computational vision (CV). First, the mathematical definition is based on the properties of a curve in the infinitesimal neighborhood about the

[6] In this paper we are primarily concerned with saliency based on *local* cues; locations on a curve where there is a transition from one type of curvature behavior to another, e.g. from perfectly straight to "wiggley," may also be psychologically salient, but such forms of *global saliency* are beyond the scope of our current investigation.

point at which curvature is being measured. For the finite precision quantized curves dealt with in CV, it has been difficult to find a suitable approximation to the limiting process originally intended for use on *mathematically continuous* curves. Second, it is readily observed that saliency is not an infinitesimal point property, but is based on some finite extent of the curve. A proposed solution to both problems, offered by Rosenfeld and Johnston [Rosenfeld73] was to find an appropriately sized segment of the curve about the point in question, and take a "snapshot" of the limiting process at this single (implied) scale. That is, rather than the rate of change of tangent angle with respect to curve length, R/J proposed measuring the angle between two fixed length chords, where the lengths correspond to the computed "natural scale" of the curve about the given point. We will call this curvature-analog the R/J-Curvature. There are a number of other definitions of mathematical curvature (e.g., the limiting radius of a circle whose three defining points converge at the curve-point in question) which have analogs that could have been used in place of the angle measure in R/J-Curvature but these definitions are monotonically related, and do not really present distinct alternatives. Thus, R/J-Curvature is a suitable representative for the whole class of mathematical curvature-measure analogs.

In Fischler and Bolles [Fischler86], our concern was not to find a good digital analog for curvature, but rather to find an effective measure of saliency. The quantity defined in that paper can be viewed as a curvature-extremum measure in which the limiting process (in scale) is replaced by a scanning process (in space) more appropriate to digital curves. The scanning process is parameterized by scale, and the resulting measure is a signed quantity which we call F/B-Saliency (F/B-S).

While the particular choice of a curvature measure as a component in a complete system for selecting the most salient points (critpts) on a planar curve depends on many factors, it is still interesting to compare the raw scores returned by *curvature-analogs represented by the R/J-Curvature* with the extreme points (ultimately) selected by our algorithm (SSS) as shown in Figures 2 and 3 for a randomly generated curve. In these figures we observe problem situations that highlight some of the differences between the two underlying metrics (R/J-Curvature and F/B-Saliency). [7]

There are some problems with *any* raw measure of curvature that must be dealt with by using procedures that invoke (at least) local *context*. For example, in Figure 3 we see a case (double arrow) where two critpts were selected at almost adjacent locations along the curve. This undesirable behavior was not eliminated by the simple "non-maximum suppression" filter that produced good results in most other situations. It is necessary to use more specific criteria in deciding when two critpts are too close together, and also, what to do when the adjacent points have equal saliency scores (e.g., arbitrarily eliminate one of them or eliminate both and place a new critpt between them). In Figure 3 we see cases (two single arrows) where almost invisible features were chosen as critpts because they *did* have locally extreme curvature scores; how do we decide when to reject such occurances. In Figure 2 we see a case where a critpt (designated by an arrow) was inserted at a location displaced from the position we consider correct; this was due, in part, to the length of the arms of the angle measuring "operator" relative to the size of the feature (see Figure 2d) – it is not always possible (or practical) to find an appropriate operator size for every potential feature. In the following sections (and appendices) of this paper we describe and justify the methods we employ to deal with these problems. The issue we are primarily concerned with in this section is the choice of a basic saliency metric. We justify our preference for the F/B-S metric on two grounds:

1. Unlike the fixed scale mathematical (FSM) curvature analogs (e.g., R/J-curvature), F/B-S rarely makes an error in positioning a critpt, or in ignoring a salient point that human observers would select. The issue here is robustness, F/B-S integrates information over an extended set of "looks" at the curve segment containing the point whose saliency is being measured. FSM techniques take a single look at the situation. Thus, our main problem with the F/B-S metric is selecting the most salient of the selected critpts to be retained as our final result (the filtering operation generally involves the elimination of less than half of the points originally selected).

2. The F/B-S metric is responsive to both the curvature and the size of a curve "feature." This provides a common basis for ranking critpts at a given scale (so that the larger of two geometrically similar objects is assigned a higher saliency score) as well as across scales by taking into account the size of the operator. The FSM-curvature analogs are insensitive to the size of the feature – they inherit the mathematical property that curvature is a point property and only the smallest neighborhood about a point that allows us to measure curvature is relevant (this implies a single "natural scale" at any point on a curve; a concept we reject, e.g., see Figure 4).

[7] In both of the figures, we used fixed common scale parameters for both metrics as noted in the figure captions. It should be remembered that R/J-curvature, as we define it in this paper, is representative of a whole class of curvature-based metrics and is not intended to duplicate the complete Rosenfeld/Johnston algorithm – they also incorporate a procedure for finding a preferred stick length. However, many of the problems with the performance of the complete algorithm, which are discussed in Davis77 and in other of the papers we reference, can be observed in the performance of the R/J-Curvature metric.

4.2 Comparison of the Saliency Selection System (SSS) with Human Performance

The primary criterion for judging the competence of the overall saliency selection system (SSS) we present in this paper is its ability to match human performance – both in the defined task and with respect to generic evaluation of the selected critpts. We performed a set of informal experiments with 11 human subjects (also see the experiments described in Fischler86). The instructions given to the subjects and the resulting selections are shown in Figure 1. We also show the selections made by the SSS algorithm. The results of these (and additional but not described) experiments can be summarized as follows

- At least 9 of the 11 subjects selected the same set of six or more critpts on each of the four curves we used in the experiments, and the SSS chose the same set of critpts. Every critpt selected by the SSS was also selected by at least one human subject.
- In spite of the high degree of consistency in the overall selection of salient points, the human subjects differed in the order in which they chose these points. We tried a number of experiments in which the only difference was a very slight change in the wording of the instructions, and obtained different orderings (across the same set of selected points) from our subjects. It is obvious that the subjects used a global strategy to match the task (different for each subject) to choose the order in which the points were selected – even though the specific points selected were largely determined by local context.

In addition to the curves used in the human experiments, we ran the SSS algorithm on (the order of) 1000 randomly generated curves with no obvious errors. Figure 5 shows the results of a (typical) sequence of 40 consecutive experiments.

5. Dealing with the Problems of Scale and Resolution

A vision system, concerned with creating a description of some object that may be encountered again in the future, perhaps when the object is closer or further away, must take scale or magnification into account when deciding what shape elements to pay attention to. Under extreme changes in resolution, when salient features might appear or disappear, it may not be possible to make an informed judgement in the assignment of relative saliency scores; but for a limited range about a given resolution, this should indeed be possible.

Obviously, geometric properties of objects that are invariant over scale are especially valuable in describing and recognizing the objects, since absolute scale is often impossible to judge in an image, and even relative scale can be difficult to describe or measure if the measurement must be referenced to the global geometry of the object. One of the main issues we address in this paper is how to define extrema in the "bending" of a curve as a local effectively scale-invariant property that is in agreement with the judgement of the human visual system.

If we define criticality of points on a digitally represented curve in terms of quantities that have dimensions that must be measured by some physical process, then there is no direct way of invoking such formally defined mathematical concepts as the derivative, or curvature, which require limiting processes of infinite resolution. Approximations to these concepts are resolution dependent (e.g., the size of the operator employed) and measurements made on most objects will not "scale" in any simple or uniform way. Further, if we examine a curve through a fixed size window (either a fixed region of a computer screen, or the foveal region of the human retina), and we successively increase the resolution at which the curve is displayed, some of its parts will eventually disappear from view, and some of the smaller original structures, that were not significant, will now dominate the visible appearance of the curve (e.g., Figure 4).

If the mathematical definition of curvature were applicable to digital imagery, then many (but not all) of the issues of scale could be resolved. There is still the problem that a very small "glitch" can have a very high value of curvature but a very low psychological significance. Thus the scale or size of a "feature" (e.g., the glitch) is an issue. The term "feature" does not appear in our problem definition; in fact, by focusing on local curve properties, we had hoped to eliminate the need to invoke this concept since an appropriate definition is far from obvious. [8] Since scale can't be ignored (even if we had a good approximation for curvature in the digital domain that was independent of scale) the following questions arise:

- The distinction, if any, between resolution and scale
- How to choose a range of scales appropriate to the specified performance criteria
- How to measure criticality at different scales
- How to compare criticality values computed at different scales
- The relationship between smoothing and scale change

[8] Intuitively, there are sections of any given curve that we call features; these entities provide the psychological basis for the selection and relative saliency of the associated critpts. Critpts are markers that define the shape and boundary of features – the extent of the curve corresponding to a feature will generally subsume the "region of support" for the curvepoints comprising the feature. Features can overlap, and their boundaries are not always apparent.

- The relation between operator size and scale change
- How to make cooperation/competition judgements across scales
- How to determine the features for which we expect consistency (of criticality scores) to hold across scales, and where such consistency can't be expected (if the latter were never the case, we could always do our analysis at one scale and compute the criticality values at other scales as needed).

While consistency at all scales and for all features is not possible, over some range of scales (say 5:1) we expect there to be a "normalization" factor which allows us to compare the saliency scores computed at one scale with values computed at other scales. We would also expect that relative locations of local extrema for *certain* features would remain fixed as a curve is scaled, regardless of the size/scale of the operator that assigns the criticality scores.

Some of the earliest work (e.g., Rosenfeld and Johnston) on finding salient points merged the problem of assigning a curvature measure to a point with that of determining the scale at which to measure curvature. The key idea is that each point has a single scale at which its curvature should be measured – this scale is usually found by a search process over successively larger scales until some measured quantity achieves a local extremum.

5.1 Change of Scale Vs. Change of Resolution

If we magnify a continuous curve that was originally represented at infinite precision, every point of the new image corresponds to a point in the original image, but its x and y coordinate values have been multiplied by some real number which we will call the scale factor. No information was introduced nor lost, but the physical space required to render the curve has increased. However, if the original curve was represented at finite resolution (e.g., each point as a pair of integer coordinates), then (say) doubling the scale leaves us with a disconnected set of points. Filling in the gaps requires introducing new information. Here we will say that a change of resolution has occurred (a change in resolution can also result in the loss of information, as in the case of demagnification or smoothing at some fixed resolution). Thus, the concept of a scale change corresponds to a reversible transformation, while, in general, a change in resolution involves an irreversible process in which information is lost (as in smoothing), or new information is introduced (as can occur in zooming).

If we compute the curvature for points on a continuous (infinite resolution) curve at two different scales, we will generally get two distinct sets of values (e.g., a circle with radius 2 is a scaled version of a circle with radius 1, but by definition, their curvatures are in the ratio 1:2. On the other hand, the angles of a triangle remain unaltered under a scale change). It will be the case, however, that for smooth curves, the local extrema will be found at corresponding locations – but even here, the numerical values of curvature will not scale in any simple way (curvature is a nonlinear function).

5.2 SSS Mechanisms for Evaluating Saliency at Different Scales and Resolutions

In designing a computational module to evaluate saliency subject to the ideas discussed above, we can pursue at least three distinct strategies:

1. Assume that saliency is independent of scale, or that there is a natural scale associated with each location on the curve that must be discovered.
2. Use a fixed scale saliency measure, but generate multiple versions of the given curve at some predetermined set of scales.
3. Parameterize the saliency measure to give results approximating those that would be obtained from strategy (2) for the selected scales.

We previously argued against strategy (1) on the assumption that a unique natural scale *cannot* generally be associated with a single curvepoint (see Figure 4). We have chosen strategy (3) since strategies (2) and (3) are conceptually compatible, but (3) could be computationally more efficient if we can find a simple way to use some combination of operator scaling and score normalization so that both approaches give (nominally) the same scores in most situations. Intuitively, doubling the stick length (in the F/B-S metric) for a simple convex section of a curve should result in four times the score assigned to the corresponding critpt: The stick is now positioned twice the distance from the critpt in most of its "looks" (i.e., placements of the stick which subsume a curve segment containing the critpt), and there are twice as many looks. Thus, the procedure we employ, *normalizing all scores by dividing by the square of the sticklength,* will leave invariant the saliency scores assigned to features which should be scale invariant, such as the angle formed by two (effectively) infinite straight lines. On the other hand, for those features that have limited extent along the curve, comparable to the scales we wish to discriminate among, the larger scaled versions of the features will be assigned higher scores.

6. Cooperation/Competition Interactions Between Critical Points

An important contribution of this paper over the work presented in Fischler and Bolles [Fischler86] is a major revision of the approach to filtering the critpts, based both on comparisons at a given scale as well as across

different scales. At a conceptual level, there are two main differences.

First, in the earlier work we did not use the information about the sign (concavity/convexity) of the computed F/B-Saliency; in our current algorithm, we separate all the candidate critpts into two sets corresponding to positive and negative F/B-S. [9] These two sets are processed independently of each other (by identical procedures) and the resulting selections are combined by logical union to produce the final output. Our own observations confirm those of other researchers (e.g., Hoffman and Richards), that positive and negative curvature extrema appear to be distinguished from each other by the HVS, in part because they play different roles in partitioning and description tasks.

Second, in the earlier work we used a simple "dominance" criterion for competition of closely spaced critpts detected at different scales.. A critpt detected at some given scale would suppress all critpts detected at smaller scales (shorter "sticklength") that were located within a specified scale related distance from it. This rule rarely produced "ugly" errors, but occasionally caused the obviously correct critpt to be deleted in favor of one slightly displaced from the preferred location. A significant portion of the work described in this paper has been focused on finding a more effective and uniform basis for establishing "local dominance." In other sections of this paper we provided a justification for a normalization factor which would permit us to assign a saliency ranking to competing critpts, regardless of the scale at which they were originally detected. Thus, competition, both within and across different scales is now treated in a uniform manner. In the following subsection we discuss some of the specific problems that must be resolved in competition resolution, and the algorithmic procedures we invoke to deal with these problems.

6.1 Mechanisms for Filtering Competing Critpts

One of the algorithmic mechanisms we devised to deal with the above problems (described in greater detail in Appendix B) is to construct an array with one slot for each indexed location along the curve (conceptually two such arrays, one each respectively for positive and negative saliency scores). Each slot is either free or "owned" by exactly one critpt. A critpt occupies only one of the slots it owns – this occupied slot corresponds to its actual location along the curve. A "new" critpt, [10] contending for a slot, must have a normalized score greater than the existing value stored in the slot to capture it. If a new critpt captures a slot occupied by (as opposed to simply being *owned* by) a previously dominant critpt, all of the slots of the now dominated critpt are also captured. This mechanism provides a way of avoiding the need to choose a fixed-sized "base of support" for a critpt.

[9] For an open curve segment, the assignment of positive vs. negative is arbitrary; the important consideration is that we use the information about the direction of deviation of the curve from the stick to separate detected critpts into the two possible categories which are then processed separately.

[10] All the potential critpts are detected, sorted, and then entered into the array in increasing order of saliency to avoid sequence dependent effects.

7. Algorithm Performance

The algorithm discussed in the previous sections of this paper, and described in Appendix B, has been compared with human performance (Figure 1), and has been run on hundreds of randomly generated images (as described in Appendix A) without making any obvious errors. In all these cases the same set of parameters were used with no operator involvement. Figure 5 shows 40 consecutively generated random curves and the critpts selected by the algorithm. Figure 6 in Appendix C shows results of the algorithm run on curves extracted from real images.

8. Discussion

Curve partitioning is an active research area which not only is of theoretical interest as a basic element in pictorial description (e.g., Attneave, Bongtsson and Eklundh, Hoffman and Richards), and for providing insight into the partitioning problem in general (e.g., Fischler and Bolles), but has many potential applications. Some of the more immediate ones include: data compression by using critpts as the basis for regenerating a curve by straight line or spline interpolation (e.g. Imai and Iri, Teh and Chin), matching/recognition using critpts and/or the partitioned curve segments (e.g., Mokhtarian and Mackworth, Wuescher and Boyer), and as a key component of an interface for man-machine communication about pictorial objects (the ability to point at icons representing symbolic objects has revolutionized the computer-user interface; to extend this capability, one would like to be able to point to a location in an image and have the machine be able to deduce the component being referred to – image partitioning in general, and especially curve partitioning, are critical to this goal).

In this paper we have focused on one specific aspect of the curve partitioning problem: Duplicating human performance in the selection of a small number of points (called *critpts*) along a curve segment which could be used as the basis for *reconstructing* the curve at some future time. While there will generally be a significant degree of overlap in the points selected by the techniques referenced above (focused on different applications), there are also significant differences. There has been very little recent work on the generic problem of choosing psychologically salient points with which to directly compare our results. On the other hand, we have conducted a relatively large number of experiments with

uniformly good results (e.g., see Figure 5).

There are two major paradigms [11] underlying the published work on partitioning planar curves. The first involves obtaining a mathematically differentiable representation of the given digital curve by the use of splining or Gaussian convolution (e.g., Mokhtarian86). This gives good results for many applications, but the salient points on the *smoothed* curve are often displaced from their original locations (or eliminated). This paradigm is not suitable for our purposes in this paper.

The second paradigm, which includes the work described here, is to first measure some approximation to the curvature at each point on a curve. This usually involves choosing, or finding, an appropriate scale at which to make the curvature measurement. This is typically accomplished by making the curvature measurement over increasingly larger curve segments (centered on the curve point being evaluated) until either the computed curvature at the point, or some related quantity, reaches a local extrema. Each point is assigned a saliency/criticality value (its estimated curvature) and an interval length along the curve centered on the point (called its *region of support*). The region of support is then used for non-maximum suppression – each point suppresses other points with lower criticality scores falling in its region of support.

Major differences between our approach and other work under this second paradigm include:

- A generic saliency measure which often selects points corresponding to local curvature extrema, but which in many situations is in better accord with human selection preference and placement accuracy.

- A distinct approach to the problem of dealing with curve *features* salient at different scales. The conventional approach is to associate a single scale with each curve point which in turn defines a fixed *region of support* to be used for non-maximum suppression. In our approach, we measure the saliency of each curve point at a number of different scales, and have developed procedures for allowing potential *critpts*, found at different scales and spatial locations to compete [12] with each other. This competition is not restricted to any fixed extent of the curve (which thus avoids anomalous selections caused by an important event occurring just beyond the fixed limit of search, i.e., the *horizon* effect).

[11] Additional approaches are available for partitioning 1-D curves; for example, see Fischler and Wolf [Fischler83] or Witkin [Witkin83]. As noted in Appendix B, the 1-D partitioning technique in the Fischler83 reference is used as a component of the SSS algorithm.

[12] It is interesting to note that we have not found a use for cooperative reinforcement – cooperation appears to be a global relation. Competition is important at the local level (e.g., lateral inhibition)

Our approach to local saliency selection can be considered a form of automated preattentive perception. Potential extensions could include dealing with more global curve features, such as recognizing the intersection of extended straight line segments, or transition points between analytic curves with different parameters, or global symmetries and repeated structure. Recognizing these more global structures, and ranking them with respect to human perceived saliency, may well fall outside the competence of the basic approach described in this paper.

9. Appendices

9.1 Appendix A: Generation of Random Curves

The following method was used to construct the random curves used in the experiments described in the body of this paper.

(1) Thirty (x,y) pairs are generated for each curve. Each value of x and y are generated by a uniform-distribution (0-1) random-number generator and then multiplied by 100 to produce numbers (coordinate-values) uniformly distributed between 0 and 100.

(2) The thirty points are next linked by a minimal-spanning-tree (MST).

(3) A diameter path is extracted from the MST, and the ordered subset of the original randomly generated points that fall along this diameter path are the input sequence provided to a spline-fitting routine [Cline74] which returns a continuous curve represented by a sequence of (x,y) coordinate pairs. These sequences, typically containing on the order of 150-250 points, are the random curves used in our experiments.

9.2 Appendix B: An Algorithm For Computing Curve-Point Criticality

The partitioning algorithm described in Fischler and Bolles [Fischler86] has been modified and extended as summarized below.

The algorithm collects candidates (peaks) for the critical points of a curve by examining the deviation of the points of the curve from a chord or "stick" that is iteratively advanced along the curve. Sticks of different lengths are used to find critical points that are salient at different "natural" scales on the given curve. (Except when explicitly stated otherwise, two sticks were used for all the experiments discussed in this paper; one of length 10 pixels and the other of length 20 pixels.) The algorithm provides the option of using arc-length along the curve, or the euclidean length of the stick, to determine the separation of the endpoints of the stick on the curve; we used the euclidean length of the stick for all of the experiments discussed in this paper. One end of the stick is advanced along the curve, one pixel at a time, and the other end is placed at the first (sequential)

position further along the curve for which the Euclidean distance equals or exceeds the specified stick length.

For each placement of the stick, an accumulator associated with the curve-point (in the interval of the curve between the two endpoints of the stick) of maximum deviation from the stick is incremented by the absolute value of the distance from the point to the stick if this distance exceeds a predefined noise threshold. However, for the given stick placement, if there is more than one excursion (exit and return) outside the noise region, the underlying model is violated and the accumulators are not incremented. (The noise threshold was uniformly set to 20 percent of stick length; thus a euclidean deviation of more than 2 "pixels" from a stick of length 10 was required to cause any modification of the associated accumulator.)

To deal with direction dependent effects, a complete traverse is made in both directions along the curve summing the results in the same accumulators. The points which have locally maximum scores in the accumulators (called peaks) for any of a given set of sticks are the points from which the critical points will be selected.

The following information is collected for each peak and used to find the critical points:

- INDEX: the sequence number along the curve of the point at which the peak was located.
- STICK: the length of the stick (in pixels) used to find the peak.
- DEV: the sign of the deviation of the peak with respect to the curve.
- NSCORE: the "normalized" score which is the score in the accumulator for the peak divided by the square of the stick length.

The peaks are divided into two groups with like-signed deviation DEV. The critical points for the two groups are found independently of each other and their union is returned as the set of critical points for the curve.

In finding the critical points, we stipulate that each peak's score has a region of support, *plus* and *minus* half its associated stick length, on each side of its position along the curve. An array (the support array) equal to the length of the curve is used to store the support information. The support information for a peak is a list (NSCORE INDEX STICK). For each peak, the support information may be entered at every index location covered by the region of support depending on what was previously stored in the location.

For all locations in the support region for the new peak (in the support array), an entry at J is replaced by the information for the new peak if there is no previous entry in the array or if the score for the new peak is > than the score in the existing entry in the array. In addition, if the entry J is being replaced, and J is also the INDEX for a peak that was entered previously, the support information for the new peak replaces the support information of the old peak wherever it occurs in the support array (i.e. even outside of the new peak's original support region).

After the above processing, the critical points for the curve are designated as those points whose index into the support array equals the index stored in the information list of the array element.

It can be seen that the order in which peaks are entered into the support array can affect the final selection of the critical points because a peak's region of support can be altered by the "capture" process, and thus depends on the state of the support array at the time the peak is entered. In our implementation of the algorithm for running the experiments, we entered the peaks into the support array as soon as they were computed in order to gain computational efficiency and simplicity, and still obtained excellent results. In the current version of the algorithm we collect all the peaks for all the sticks, sort the peaks by their normalized scores, and then enter them into the support array in order of increasing score.

There are some additional aspects of the algorithm that are further discussed in the more complete version of this paper, including ways to handle problems associated with very sharp angles and competing critpts of approximately equal saliency scores,

9.3 Appendix C: Partitioning Curves Extracted From Aerial Imagery

A technique for detecting and delineating low resolution linear structures appearing in aerial imagery, such as roads and rivers, was described by the authors of this paper in an earlier publication [Fischler83]. The algorithm was effective in finding such structure, but it provided no mechanism for distinguishing between the semantically meaningful objects and the "accidental" and irrelevant linear features found in most real images. In work now in progress, we use the SSS algorithm to "slice up" the individual curves found by the delineation algorithm. We throw away the very small resulting segments which are typical of accidental linear formations, and then further filter the longer segments with respect to a set of semantic constraints. Those segments that pass through the filtering process are then "glued" back together to produce the desired delineation. This process is illustrated in Figure 6. Figure 6a shows an aerial image, and 6b shows the linear segments extracted by use of the original delineation algorithm. Figure 6c shows those segments that passed through the filters mentioned above, and Figure 6d shows the result of a final step to retain only the more significant roads and trails. The two panes of Figure 6e show the results of applying the SSS algorithm to some of the 120 curves highlighted in Figure 6b (they have been isolated and separated into the two panes to allow clear display of the partition points and

to prevent confusion due to the intersections of distinct curves). The robustness of the SSS algoritmn is essential in carrying out the filtering operation. Insertion of extraneous partition points would cause the loss of portions of the road network; absence of valid partition points would allow meaningless appendages to become part of the extracted network.

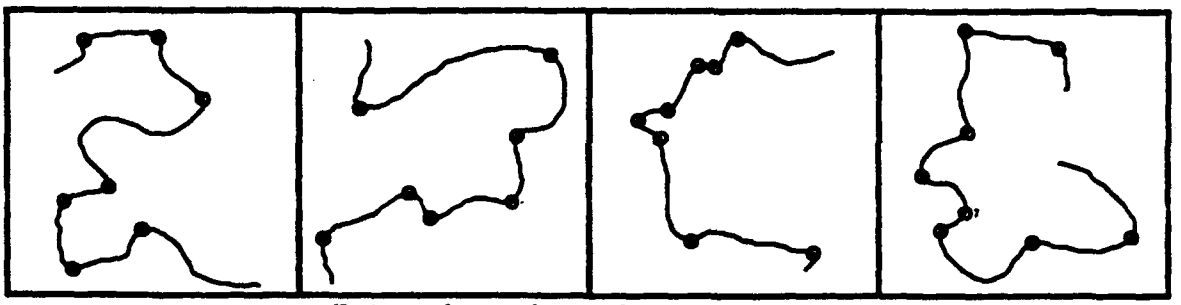
10. References

1. F. Attneave, "Some informational aspects of visual perception, "Psychol. Rev. 61:183-193, 1954.
2. A. Bengtsson and J.O. Eklundh, "Shape Representation by Multiscale Contour Approximation," IEEE Trans PAMI-13(1):85-93, Jan. 1991.
3. A.K. Cline, "Scalar- and planar- valued curve fitting using splines under tension," CACM 17(4):218-223, April 1974.
4. L.S. Davis, "Understanding shape: angles and sides," IEEE Trans. Comput. C-26:236-242, March 1977.
5. M.A. Fischler and R.C. Bolles, "Perceptual organization and curve partitioning," IEEE Trans PAMI-8(1):100-105, Jan. 1986.
6. M.A. Fischler and H.C. Wolf, "Linear Delineation," Proceedings IEEE CVPR-83, June 1983, pp 351-356; also, Readings in Computer Vision (M.A. Fischler and O. firschein, eds.), Morgan Kaufmann, pp 204-209, 1987.
7. M.A. Fischler and P. Barrett, "An iconic transform for sketch completion and shape abstraction," CGIP 13:334-360, 1980.
8. D. Hilbert and S. Cohen-Vossen, "Geometry and the imagination." Chelsea, 1952.
9. D.D. Hoffman and W.A. Richards, "Representing smooth plane curves for recognition: implications for figure-ground reversal," Proc. 2nd Nat. Conf. Artificial Intelligence, Pittsburg, PA, pp 5-8, Aug. 1982.
10. H. Imai and M. Iri, "Computational-geometric methods for polygonal approximations of a curve," CVGIP-36(1):31-34, Oct. 1986.
11. D.G. Lowe, "Organization of smooth image curves at multiple scales," Proc 2nd ICCV, pp. 558-567, 1988.
12. R. Mehrotra, S. Nichani, and N. Ranganathan, "Corner detection," Pattern Recognition 23(11):1223-1233, 1990.
13. F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," IEEE PAMI 8(1):34-43, Jan 1986.
14. T. Pavlidis and S.L. Horowitz, "Segmentation of plane curves," IEEE Trans. Comput. C-23:860-870, Aug. 1974.
15. W. Richards and D. Hoffman, "Codon constraints on closed 2D shapes," in Human and Machine Vision II (A. Rosenfeld, ed.), Academic Press, pp 207-223, 1986.
16. W. Richards, B. Dawson,, and D. Whittington, "J. Optical Soc. Amer. 3(9):1483-1491, Sept. 1986.
17. A. Rosenfeld and E. Johnston, "Angle detection in digital curves," IEEE Trans. Comput. C-22:875-878, 1973.
18. A. Rosenfeld and J.S. Weszka, "An improved method of angle detection on digital curves." IEEE Trans. Comput. C-24:940-941, Sept. 1975.
19. C.H. Teh and R.T. Chin, "On the detection of dominant points on digital curves," IEEE Trans PAMI-11(8):859-872, Aug. 1989.
20. A. Witkin, "Scale Space Filtering," Proc. 8th IJCAI, Karlsruhe, West Germany, pp 1019-1022, Aug. 1983.
21. D.M. Wuescher and K.L. Boyer, "Robust contour decomposition using a constant curvature criterion," IEEE Trans PAMI-13(1):41-51 Jan. 1991.

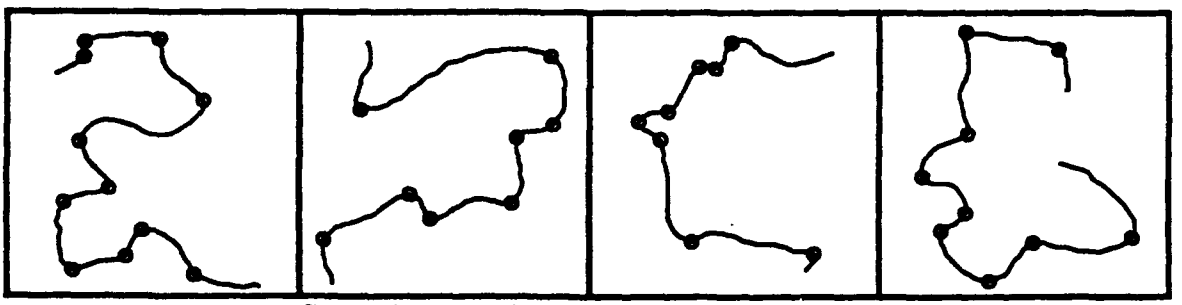
CURVE PARTITIONING: Instructions

For each enclosed curve:

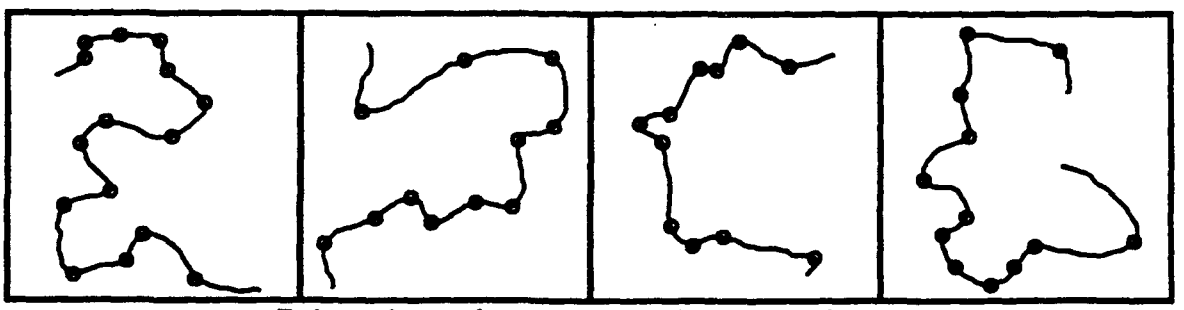
Assume that 10 years from now you will be asked to reconstruct the given curve. A reasonably correct reconstruction will be rewarded by a large sum of money (say $5000). You can record, for later use, the locations of up to nine points along the curve to help you do the reconstruction – but it will cost you $200 for each such point (to be subtracted from your prize if you receive the reward). Please mark your selected points on the curve. Do not select the endpoints, they will be provided free. Do not take more than one minute per curve.



Points chosen by 9 of 11 test subjects

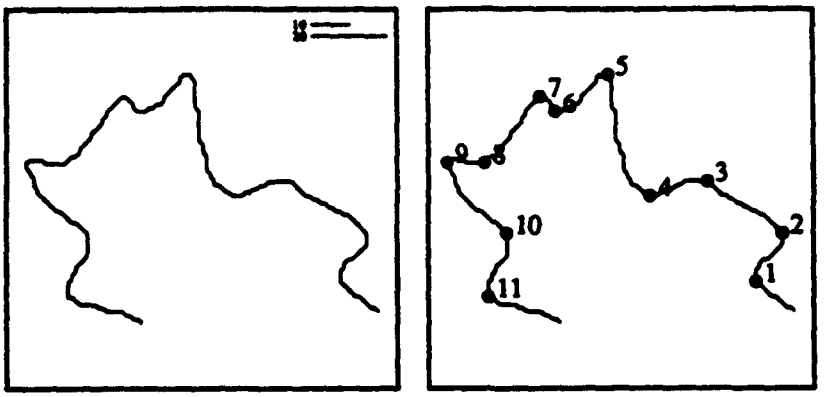


Critical points found by the SSS algorithm



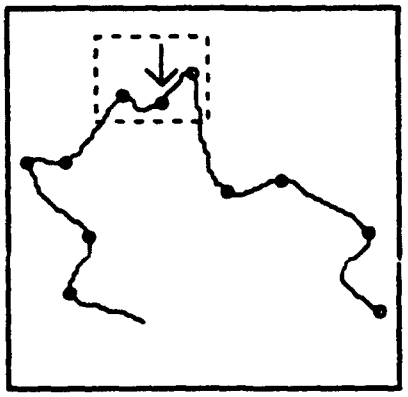
Points chosen by at least 1 of 11 test subjects

Figure 1: Comparison of human and SSS algorithm performance in the curve partitioning task. (Each of the curves used in the experiments with human subjects was contained in a square that was 1.5 inches on a side.)

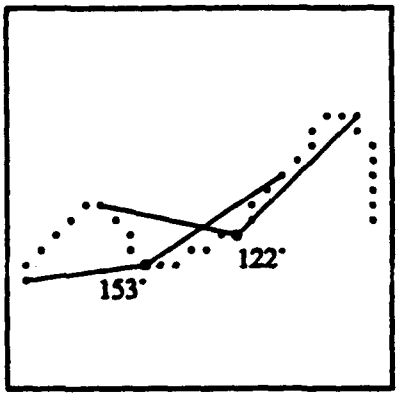


(a) Test curve 189

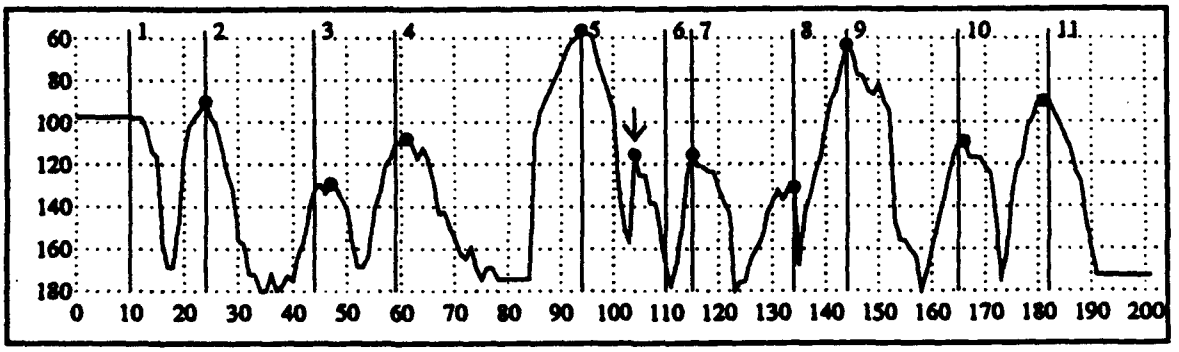
(b) SSS selected critpts



(c) R/J-curvature selected critpts

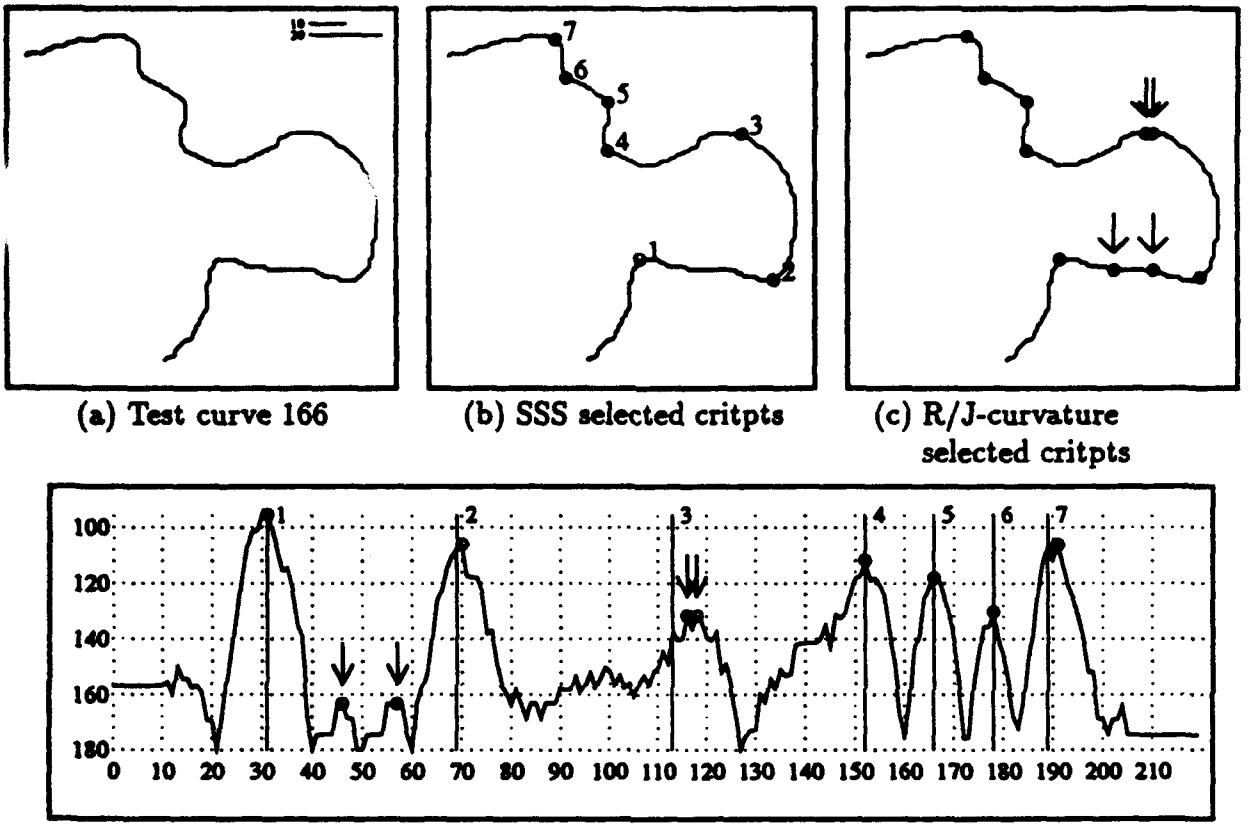


(d) Anomalous area (magnified)



(e) Plot of R/J-curvature along test curve. Abcissa = sequence number of point on curve. Ordinate = angle (in degrees) computed at point. (Angle-arms are 10 units each for R/J-C; standard stick lengths of 10 and 20 units are employed by SSS.)

Figure 2: Comparison of SSS and R/J-curvature metrics evaluated on test curve 189. The continuous curve in (e) represents R/J-curvature along the test curve shown in (a). The vertical lines in (e) mark the sequentially numbered critpts selected by SSS as shown in (b). The critpts corresponding to the extreme values of R/J-curvature shown in (c) are marked as circles in (e). The arrow in (c), and in the corresponding location in (e), illustrates an anomalous selection using R/J-curvature. (d) shows the computed values of R/J-curvature, 153°, at the preferred location and 122° at the location of the anomalous selection.



(a) Test curve 166 (b) SSS selected critpts (c) R/J-curvature selected critpts

(d) Plot of R/J-curvature along test curve. Abcissa = sequence number of point on curve. Ordinate = angle (in degrees) computed at point. (Angle-arms are 10 units each for R/J-C; stick length is 20 units for F/B-S.)

Figure 3: Comparison of SSS and R/J-curvature metrics evaluated on test curve 166. The continuous curve in (d) represents R/J-curvature along the test curve shown in (a). The vertical lines in (d) mark the sequentially numbered critpts selected by SSS as shown in (b). The critpts corresponding to the extreme values of R/J-curvature shown in (c) are marked as circles in (d). The arrows in (c), and in the corresponding locations in (d), illustrates anomalous selections using R/J-curvature.

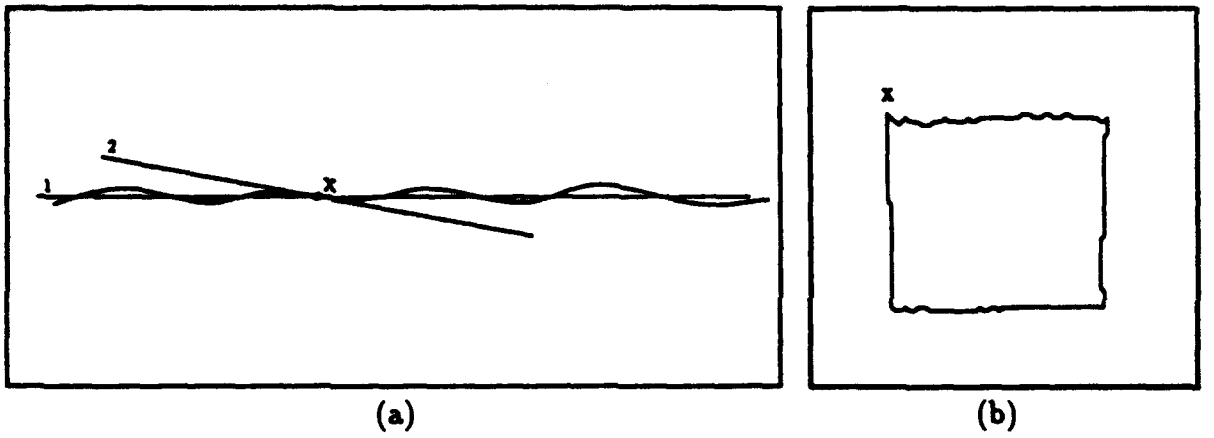


Figure 4: Curvature and saliency are functions of curve resolution. As illustrated in (a) above, we can draw more than one visually acceptable tangent to many of the points on this curve at the given resolution. As resolution increases, tangent 2 would dominate at point x; as resolution decreases, tangent 1 would dominate at the same point. In (b), the angle at x can be seen as 45° at one scale and 90° at a larger scale. Thus, curvature and saliency are not unique properties of curve points.

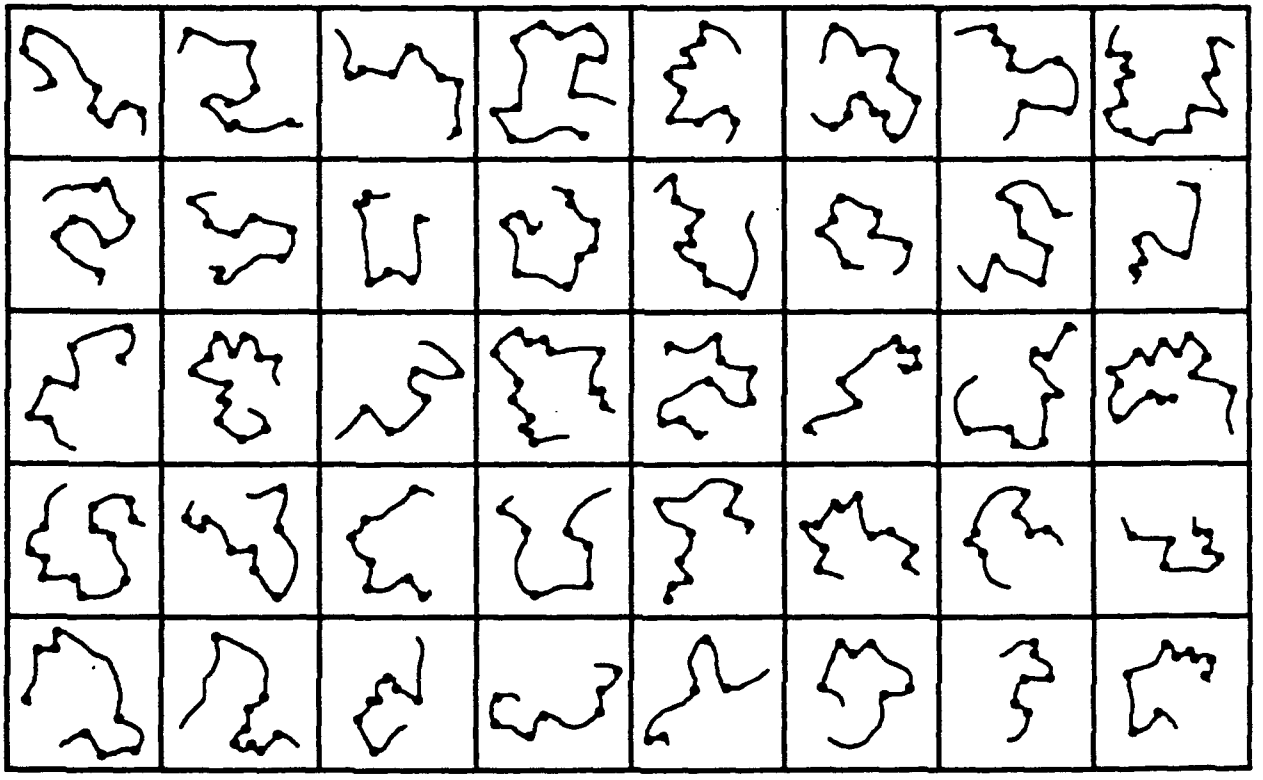
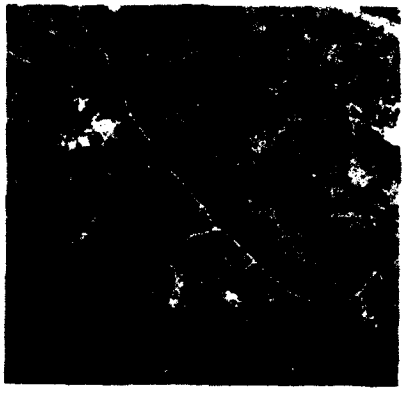
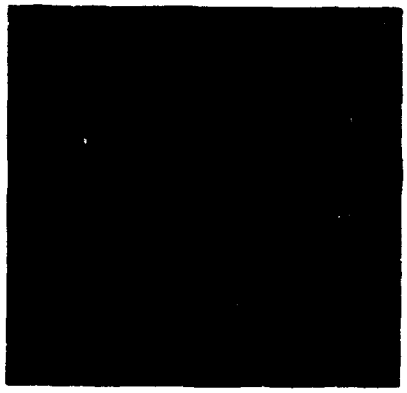


Figure 5: Critical points found by the SSS algorithm for a set of 40 random curves.

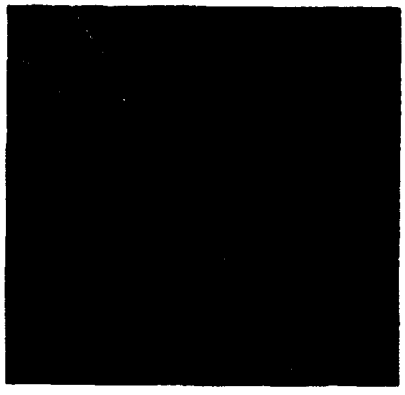


(a) Aerial photograph

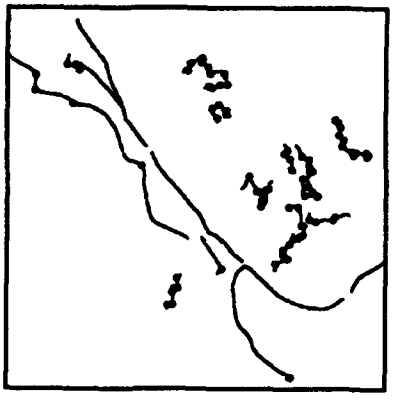


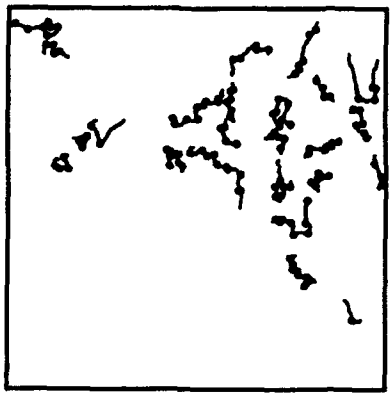
(b) Initial extraction of linear structure

(c) Filtered linear structure using SSS algorithm



(d) Delineation of major roads and trails





(e) Partition points found by SSS algorithm on curves from (b)

Figure 6: Application of the SSS algorithm to the problem of delineating linear features in aerial photographs.

Appendix C:

Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading

Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading*

P. Fua and Y. Leclerc
SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025
(fua@ai.sri.com leclerc@ai.sri.com)

Abstract

Our goal is to reconstruct both the shape and reflectance properties of surfaces from multiple images. We argue that an object-centered representation is most appropriate for this purpose because it naturally accomodates multiple sources of data, multiple images (including motion sequences of a rigid object), and self-occlusions. We then present a specific object-centered reconstruction method and its implementation. The method begins with an initial estimate of surface shape (provided by triangulating the result of conventional stereo or other means). The surface shape and reflectance properties are then iteratively adjusted to minimize an objective function that combines information from multiple input images. The objective function is a weighted sum of "stereo," shading, and smoothness components, where the weight varies over the surface. For example, the stereo component is weighted more strongly where the surface projects onto highly textured areas in the images, and less strongly otherwise. Thus, each component has its greatest influence where its accuracy is likely to be greatest. Experimental results on both synthetic and real images are presented.

1 Introduction

The problem of recovering the shape and reflectance properties of a surface from multiple images has received considerable attention [6, 20, 35, 44]. This is a key problem not only in developing general-purpose vision systems, but also in specialized areas such as the generation of Digital Elevation Models from aerial images [5, 12, 26, 53].

In this paper, we view the recovery problem as one of finding an object-centered description of a surface from a set of input images that is sufficiently complete, in terms of its geometric and radiometric properties, that it is possible to generate an image of the surface from any viewpoint. In particular, the description should be sufficiently complete to reproduce the input images to within a certain tolerance, given models of the cameras, their relative locations, and expected noise.

Our surface reconstruction method uses an object-centered representation, specifically, a triangulated 3-D mesh of vertices. Such a representation accommodates the two classes of information mentioned above, as well as multiple images (including motion sequences of a rigid object) and self-occlusions. We have chosen to model the surface material using the Lambertian reflectance model with variable albedo, though generalizations to specular surfaces would be relatively straightforward. Consequently, the natural choice for the monocular information source is shading, while intensity is the natural choice for the image feature used in multi-image correspondence. Not only are these the natural choices given a Lambertian reflectance model, they are also complementary [7, 30]: intensity correlation is most accurate wherever the input images are highly textured, whereas shading is most accurate when the input images are untextured.

*Support for this research was provided in part by various contracts from the Advanced Research Projects Agency.

The reconstruction method is to minimize an objective function whose components depend on the input images and some measure of the complexity of the 3–D mesh. The method starts with an initial estimate for the mesh derived from the triangulation of conventional stereo results, and uses a standard optimization technique called *conjugate gradient descent* to minimize the objective function. The image-dependent components of the objective function are related to the two sources of information mentioned above. We take advantage of the complementary nature of the information sources by weighting the components at each facet of the triangulated mesh according to the amount of texturing within the area of the images that the facet projects to. The projection uses a hidden-surface algorithm to take occlusions into account.

In the following section, we describe related work and our contributions in this area. Following this we discuss some of the key issues in multi-image surface reconstruction and how to combine different sources of information for such purposes. We then describe in detail our specific procedure, discuss the behavior of our procedure on synthetic data, and show some results on real images.

2 Related Work and Contributions

Three-dimensional reconstruction of visible surfaces continues to be an important goal of the computer vision research community. Initially, much of the work concentrated on $2\frac{1}{2}$–dimensional image-centered reconstructions, such as Barrow and Tenenbaum's *Intrinsic Images* [6] and Marr's $2\frac{1}{2}$*–D Sketch* [35]. These preliminary ideas have been the basis for quite successful systems for recovering shape and surface properties. Some have used single sources of information, such as sequences of range data or intensity images [3, 25], stereo [12, 26, 52, 53], and shading [21, 24, 44]. Others have combined sources of information, such as shading and texture [8], features and stereo [23], focus, vergence, stereo, and camera calibration [1]. See [2] for further discussions on information fusion.

More recently, full 3-dimensional models have been used, such as 3–D surface meshes [46, 49], parameterized surfaces [40, 33], particle systems [42, 17], and volumetric models [36, 45, 37].

As with the $2\frac{1}{2}$–dimensional representations, 3–D representations have used a variety of single image cues for reconstruction, such as silhouettes and image features [9, 11, 47, 48, 50], range data [51], stereo [17], and motion [41]. Liedtke[32] first uses silhouettes to derive an initial estimate of the surface, and then uses a multi-image stereo algorithm to improve on the result. Their approach to deriving an initial estimate for the mesh, as with Szeliski and Tonneson's approach [42], is significantly more powerful than the one we use in this paper. This is an important topic for future research.

Of special relevance to this paper is research in combining stereo and shape from shading. Using $2\frac{1}{2}$–dimensional representations, Blake *et al.* [7] is the earliest reference we are aware of that discusses the complementary nature of stereo and shape from shading, but their experimental results are almost non-existent in this paper. Leclerc and Bobick [31] discuss the integration of stereo and shape from shading, but only use stereo as an initial condition to a discrete height from shading algorithm. Cryer *et al.* [10] combine the high-frequency information from a shape from shading algorithm with the low-frequency information from a stereo algorithm using filters designed to match those in the human visual system.

Using full 3–D representations, Heipke [22] integrates the two cues, but assumes that the images can be separated beforehand into zones of variable albedo (where one does stereo) and areas of constant albedo (where one does shape from shading). This is in contrast to our approach, in which the optimization procedure dynamically adapts to the image data.

In this paper, we unify the idea of using 3–D meshes to integrate information from multiple images with that of using multiple cues. Our specific approach to this unification, has led to a number of important contributions:

- We correctly deal with occlusions by using a hidden surface algorithm during the re-

construction process.

- Our technique for doing stereo avoids the constant depth assumption of traditional correlation-based stereo algorithms, effectively using variable-sized windows in the images.
- Our approach to shape from shading is applicable to surfaces with slowly varying albedo. This is a significant advance over traditional approaches that require constant albedo.
- We propose a dynamic weighting scheme for combining shape from shading and stereo, and demonstrate that it leads to significantly better results than using either cue alone using both synthetic and real images.

To demonstrate the validity of the overall approach, we have implemented a computationally effective optimization procedure, and have demonstrated that it finds good minima of the objective function on both synthetic and real images.

3 Issues in Multi-Image Surface Reconstruction

In this section, we briefly discuss some of the key issues in multi-image surface reconstructions, and outline how we address the issues in this paper. These outlines will be expanded upon in Section 4.

3.1 Surface Shape and its Representation

Since the task is to reconstruct a surface from multiple images whose vantage points may be very different, we need a surface representation that can be used to generate images of the surface from arbitrary viewpoints, taking into account self-occlusion, self-shadowing, and other viewpoint-dependent effects. Clearly, a single image-centered representation, such as a depth map, is inadequate for this purpose. Instead, an object-centered surface representation is required.

There are many object-centered surface representations that are possible. However, there are some practical issues that are important in choosing an appropriate one. First, the representation should be general-purpose in the sense that it should be possible to represent any continuous surface, closed or open, and of arbitrary genus. Second, it should be relatively straightforward to generate an instance of a surface from standard data sets such as depth maps or clouds of points. Finally, there should be a computationally simple correspondence between the parameters specifying the surface and the actual 3-D shape of the surface, so that images of the surface can be easily generated, thereby allowing the integration of information from multiple images.

A hexagonally connected mesh of 3-D vertices, as in Figure 2, is an example of a surface representation that meets the criteria stated above, and is the one we have chosen for this paper. Such a mesh defines a surface composed of three-sided planar polygons that we call triangular facets, or simply facets. Triangular facets are particularly easy to manipulate for image and shadow generation, since they are the basis for many 3-D graphics systems. Hexagonal meshes can be used to construct virtually arbitrary surfaces. Finally, standard triangulation algorithms can be used to generate such a surface representation from real noisy data [18, 42].

3.2 Material Properties and their Representation

Objects in the world are composed of many types of material, and the material type can vary across the object's surface in many ways. The key issues, therefore, are the type of material we wish to consider, and how its variation across the surface is to be represented. In general, one can represent a material type by its reflectance function, which maps the wavelength distribution and orientation of a light source, the normal to the surface, and the viewing direction into an image color. This function is generally quite complex. However, there are reflectance functions that are not only much simpler, but are also quite common. Such functions are modeled using only one, or, at most, a few,

parameters. Consequently, one can accurately model the material properties of a surface by representing these parameters at every point on the surface.

Probably the simplest, and most common, such function is the Lambertian reflectance function. For grey-level images, this function not only has a single parameter, albedo, which is the ratio of incoming to outgoing light intensity, but the image intensity is independent of viewpoint. For this reason, we have chosen to restrict ourselves to Lambertian surfaces in this paper. However, because we use a full 3-D representation, a generalization to specular surfaces would be fairly straightforward.

Having chosen a specific reflectance function, the remaining issue is how to represent the spatially-varying parameter(s). In general, one needs to be able to represent independent parameter values at every point of the surface. In terms of the mesh representation of the surface, this implies some type of spatial sampling of each facet. Given the finite resolution of the images, and other practical considerations, we have chosen to use two types of spatial sampling. The first is most appropriate when the parameters vary quickly across the surface, and the second when they vary more slowly. For the former case, we use a uniform sampling of each facet, where the inter-sample spacing corresponds roughly to no more than one or two pixels in any of the images. For the later case, we use a single value associated with each facet.

As we shall see later, the two different representations are used somewhat differently, and the choice of which representation to use is made on a facet-by-facet basis as a function of the images.

3.3 Information Sources for Reconstruction

There are a number of information sources that are available for the reconstruction of a surface and its material properties. Here, we consider two classes of information.

The first class are those information sources that require a single image, such as texture gradients, shading, and occlusion edges. When using multiple images and a full 3-D surface representation, however, we can do certain things that cannot be done with a single image. First, the information source can be checked for consistency across all images, taking into account occlusions. Second, the information can be "averaged" over all the images, when the source is consistent and occlusions are taken into account, to increase its sensitivity.

The second class are those information sources that require at least two images, such as the triangulation of corresponding points between input images (given camera models and their relative positions). Generally speaking, this source is most useful when corresponding points can be easily identified, and their image positions accurately measured. The ease and accuracy of this correspondence can vary significantly from place to place in the image set, and depends critically on the type of feature used. Consequently, whatever the type of feature used, one must be able to identify where in the images that feature provides reliable correspondences, and what accuracy one can expect.

The image feature that we have chosen for correspondence (though it is by no means the only one possible) is simply intensity, because the Lambertian reflectance model described earlier implies that the image intensity of a surface point is independent of the viewing direction. Therefore, corresponding points should have the same intensity in all images. Clearly, intensity can only be a reliable feature when the albedo varies quickly enough on the surface (and, consequently, the images are highly textured), and the search space is sufficiently narrow. Otherwise, there would be significant ambiguity in the correspondence of pixels across the images.

In contrast to our approach traditional correlation-based stereo methods use fixed-size windows in images, which can only yield correct results when the surface is tangential to the image plane. Instead, we compare the intensities as projected onto the facets of the surface, which is equivalent to having variable-shaped windows in the images. Consequently, if the original surface is well modeled by a mesh surface, the reconstruction can be significantly more accurate. The Hierarchical Warp Stereo System [39] is another example of a method that takes into account the variable shapes of windows required

for accurate reconstruction of a surface, though it uses only an image-centered representation of the surface.

As for the monocular information source, we have chosen to use shading. There are a number of reasons for this. First, we are using a Lambertian reflectance model, making shading a relatively simple source of information. Second, shading is most reliable when the albedo varies slowly across the surface, which is the natural complement to intensity correspondence, which requires quickly varying albedo. The complementary nature of these two sources should allow us to accurately recover the surface geometry and material properties for a wide variety of images.

In contrast to our approach traditional uses of shading information assume that the albedo is constant across the entire surface, which is a major limitation when applied to real images. We overcome this limitation by improving upon a method to deal with discontinuities in albedo alluded to in the summary of [30, 31]. We compute the albedo at each facet using the normal to the facet, a light-source direction, and the average of the intensities projected onto the facet from all images. Since we use the average of the projected intensities, this computed albedo minimizes the mean squared error between the images of the mesh surface and the input images. The variation of this computed albedo across the surface is the actual information source used to recover the surface. For example, if the albedo of the real surface were indeed constant, as in traditional shape-from-shading problems, then the measured variation in albedo will be zero for the correct mesh surface, and we will have recovered both surface shape and albedo. The distinct advantage of this approach over the traditional one is that it can deal with surfaces whose albedo is not constant, but instead varies slowly over the surface.

In the following subsection, we describe how these two sources of information are combined and used to reconstruct surfaces.

3.4 Combining and Using Information Sources

Simply put, our approach to surface reconstruction is to adjust the parameters of the surface (in the case of the mesh, this means the coordinates of the vertices), until the images of the surface are most consistent with the information sources described above. This approach requires a number of things. First, one must have an initial estimate of the surface. In this paper, this is derived from a standard correlation-based stereo algorithm. Second, one must know the light source direction, camera models, and their relative positions so that images of the surface can be generated (we assume these are provided a priori). Third, one must have a way of quantifying what is meant by "most consistent with the information sources." Here, we use an objective function that is a linear combination of components, one for each information source, whose weights are determined on a facet-by-facet basis as a function of the images. Finally, one must have a computationally effective means of finding a surface, given the initial estimate, that is reasonably close to the best of all possible surfaces according to the objective function.

Our combined objective function has three components, two of which were mentioned above: an intensity correlation component, and an albedo variation component. A third component is a measure of the smoothness of the surface. The first two components are weighted differently at each facet as a function of the image intensities projected onto the facet, while the surface smoothness component has the same weight everywhere, but is typically decreased as the iterations proceed.

Since the intensity correlation component depends on the difference in intensity at a given point, it is most accurate when the images are highly textured in the areas that the facet projects to. To see this, consider the case when the images have constant intensity in the neighborhood of the projected facet: the difference in intensity will be a constant, independent of small variations in the facet's position or orientation. On the other hand, when the images are highly textured, small changes in the facet

can significantly change the value of this component. Thus, we weight the intensity correlation component most strongly for those facets in which the projected image intensities are highly textured.

Conversely, the albedo variation component is most accurate when the intensities within a facet vary slowly. This is because we are assuming that the albedo varies slowly enough across the surface that a constant-albedo facet is a good model for the surface. Since the facets are planar, this should produce images whose intensities are constant within the projected facet. Thus, we weight the albedo variation component most strongly when the projected intensities within a facet vary slowly.

Since rapidly changing albedoes produce highly textured image regions, our weighting scheme, in effect, turns off the shading component and turns on the stereo component in such regions. Thus, it provides the shape from shading component with implicit boundary conditions at the edge of regions of constant albedo.

The surface smoothness component is required as a stabilizing term because neither of the above components is likely to be exactly correct, the surfaces are not exactly Lambertian, the camera positions are not exactly correct, there is noise in the images, and so on. Currently, we use the heuristic technique of starting with a relatively large weight for the smoothness component, and decrease it as the iterations proceed. The theoretically optimal point at which the smoothness weight should no longer be decreased is still an open question, although a single, empirically determined, value has been used with great success across all of the images presented in this paper when using all of the components.

In the following section, we describe the surface representation and optimization algorithm in more detail.

4 Details of Surface Model and Optimization Procedure

As discussed in the previous section, our approach to recovering surface shape and reflectance properties from multiple images is to deform a three-dimensional representation of the surface so as to minimize an objective function. The free variables of this objective function are the coordinates of the vertices of the mesh representing the surface, and the process is started with an initial estimate of the surface. For the experiments described in this paper, we have derived this initial estimate by triangulating the smooth depth-map generated by the correlation-based stereo algorithm described in [19, 15]. Figure 1 illustrates the output of this algorithm on a synthetic stereo pair.

Alternatively, we could have relied on more sophisticated algorithms that can triangulate noisy laser or stereo range-data to derive our initial estimates [14, 18, 42]. All these methods tend to smooth the data and to interpolate blindly in the absence of data so that their output needs to be refined by algorithms such as ours.

In this section, we describe more formally each part of our approach.

4.1 Images and Camera Models

In this paper, we assume that images are monochrome, and that their camera models are known *a priori*. The set of grey-level images is denoted $\mathbf{G} = (g_1, g_2, \ldots, g_{n_g})$. A point in an image is denoted $\mathbf{u} = (u, v)$, and the intensity of point $\mathbf{u}$ in image g_i is denoted $g_i(\mathbf{u})$. For non-integer values of $\mathbf{u}$ we use bilinear interpolation over the four points represented by the floor and ceiling of the coordinates of $\mathbf{u}$.

The projection of an arbitrary point $\mathbf{x} = (x, y, z)$ in space into image g_i is denoted $\mathbf{m}_i(\mathbf{x})$. There are well-known methods for correcting both geometric and radiometric errors in images, as surveyed in [4], pp. 68–77. Thus, we assume that all effects of lens distortion and the like have been taken care of in producing the input images, so that the projection of a surface into an image is well modeled by a perspective projection. Thus, $\mathbf{u} = \mathbf{m}_i(\mathbf{x})$ can be written as:

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = M_i \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figure 1: (a,b) A synthetic stereo pair generated by texture-mapping a real image of the Martin-Marietta ALV test-site onto a Digital Elevation Model (DEM). (c) The disparity map using a correlation-based algorithm. The black areas indicate that the stereo algorithm could not find a match. Elsewhere, lighter greys indicate higher elevations. (d) The same disparity map after smoothing and interpolation.

$$u = U/W$$
$$v = V/W,$$

where M_i is a three by four projection matrix.

4.2 Surface Representation

We represent a surface $\mathcal{S}$ by a hexagonally-connected set of vertices $\mathbf{V} = (v_1, v_2, \ldots, v_{n_v})$ called a *mesh*. The position of vertex v_j is specified by its Cartesian coordinates (x_j, y_j, z_j). Figure 2 shows such a mesh as a wire frame and as a shaded solid surface.

Each vertex in the interior of the surface has exactly six neighbors. The neighbors of vertex v_j are consistently ordered in a clock-wise fashion. Vertices on the edge of a surface may have anywhere from two to five neighbors.

Neighboring vertices are further organized into triangular planar surface elements called *facets*, denoted $\mathbf{F} = (f_1, f_2, \ldots, f_{n_f})$. The vertices of a facet are also ordered in a clock-wise fashion. In this work, we require that the initial estimate of the surface have facets whose sides are of equal length. The objective function described below tends to maintain this equality, but does not strictly enforce it. The representation can be extended in a straight-forward fashion to support different surface resolutions by sub-dividing facets (which we have done). However, facets of a given resolution will still be required to have approximately equal sides.

4.3 Objective Function

The objective function $\mathcal{E}(\mathcal{S})$ that we use to recover the surface is best described in two equations. In the first equation,

$$\mathcal{E}(\mathcal{S}) = \lambda_D \mathcal{E}_D(\mathcal{S}) + \mathcal{E}_G(\mathcal{S}), \qquad (1)$$

$\mathcal{E}(\mathcal{S})$ is decomposed into a linear combination of two components. The first component, $\mathcal{E}_D(\mathcal{S})$, is a measure of the deformation of the surface from a nominal shape, and is independent of the images. For this paper, the nominal shape is a plane. Higher-order measures, such as deformation from a sphere, are also possible. This nominal shape represents the shape that the surface would take in the absence of any information from the images.

The second component,

$$\mathcal{E}_G(\mathcal{S}) = \lambda_C \mathcal{E}_C(\mathcal{S}) + \lambda_S \mathcal{E}_S(\mathcal{S}) \qquad (2)$$

depends on the images, and is the one that drives the reconstruction process. It is further decomposed into a linear combination of the two information sources described in the previous section: a multi-image correlation component, $\mathcal{E}_C(\mathcal{S})$, and a component that depends on the shading of the surface, $\mathcal{E}_S(\mathcal{S})$.

These components, and their relative weights, are described in more detail below.

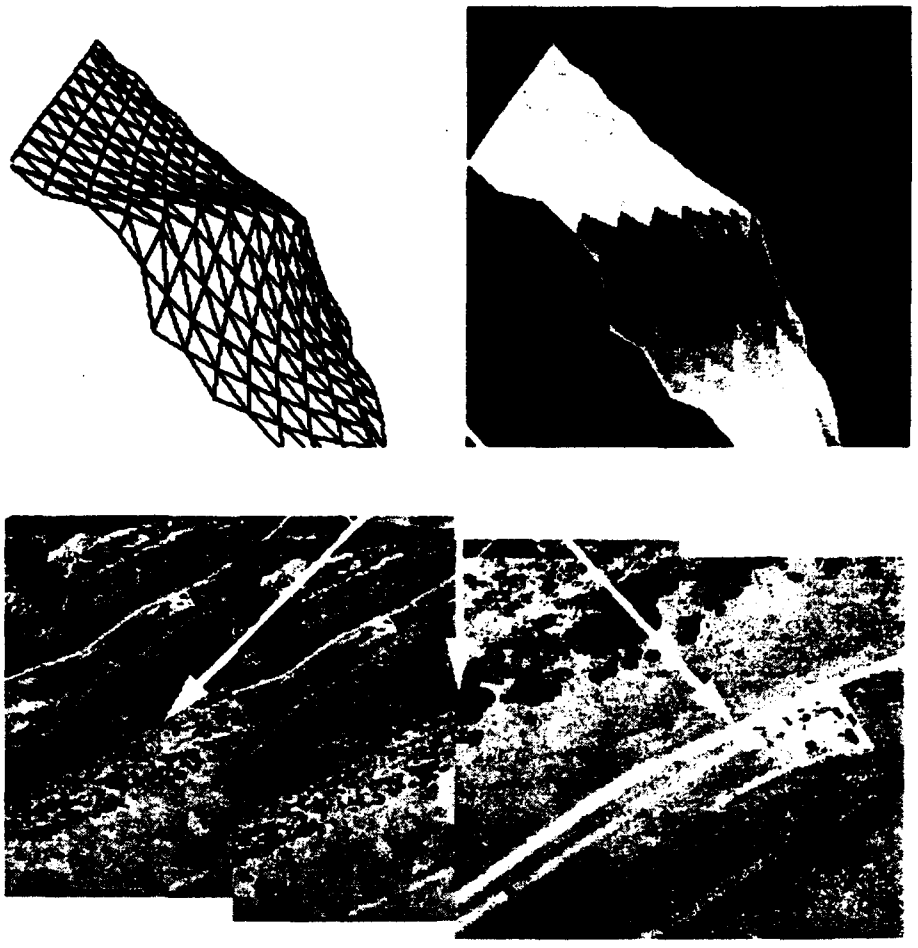


Figure 2: The top row shows a hexagonal mesh as both a wireframe and a shaded surface. The bottom row shows several images of a scene. In our approach, these images are projected onto the mesh using camera models.

4.3.1 Surface Deformation Component

As stated earlier, the surface deformation (or smoothness) component is a measure of the deviation of the mesh surface from some nominal smooth shape. When the nominal shape is a plane, we can approximate this as follows.

Consider a perfectly planar hexagonal mesh for which the distances between neighboring vertices are exactly equal. Recall that the mesh is defined so that the neighbors of a vertex v_i are ordered in a clock-wise fashion, and are denoted $v_{N_i(j)}$. If the hexagonal mesh was perfectly planar, then the third neighbor over from the j^{th} neighbor, $v_{N_i(j+3)}$, would lie on a straight line with v_i and $v_{N_i(j)}$. Given that the inter-vertex distances are equal, this implies that coordinates of v_i equal the average of the coordinates of $v_{N_i(j)}$ and $v_{N_i(j+3)}$, for any j.

Given the above, we can write a measure of the deviation of the mesh from a plane as follows:

$$\mathcal{E}_D(\mathcal{S}) = \sum_{i=1}^{n_v} \sum_{\substack{j=1 \\ k=N_i(j) \\ k'=N_i(j+3)}}^{3} \begin{matrix} (2x_i - x_k - x_{k'})^2 + \\ (2y_i - y_k - y_{k'})^2 + \\ (2z_i - z_k - z_{k'})^2 \end{matrix} .$$

Note that this term is also equivalent to the squared directional curvature of the surface when the sides have approximately equal lengths [27]. Also, this term can accommodate multiple resolutions of facets by normalizing each term by the nominal inter-vertex spac-

ing of the facets.

4.3.2 Multi-Image Intensity Correlation

The multi-image intensity correlation component is the sum of squared differences in intensity from all the images at a given sample-point on a facet, summed over all sample-points, and summed over all facets. This component is presented in stages in the remainder of this subsection.

First, we define the sample-points of a facet by taking advantage of the fact that all points on a triangular facet are a convex combination of its vertices. Thus, we can write the sample-points $\mathbf{x}_{k,l}$ of facet f_k as:

$$\mathbf{x}_{k,l} = \lambda_{l,1}\,\mathbf{x}_{k,1} + \lambda_{l,2}\,\mathbf{x}_{k,2} + \lambda_{l,3}\,\mathbf{x}_{k,3},\ l = 3, 4, \ldots n_s,$$

where $\mathbf{x}_{k,1}$, $\mathbf{x}_{k,2}$, and $\mathbf{x}_{k,3}$ are the coordinates of the vertices of facet f_k, and $\lambda_{l,1} + \lambda_{l,2} + \lambda_{l,3} = 1$. In the top half of Figure 3(a), we see an example of the sample points of a facet.

Next, we develop the sum of squared differences in intensity from all images for a given point $\mathbf{x}$. Recall that a point $\mathbf{x}$ in space is projected into a point $\mathbf{u}$ in image g_i via the perspective transformation $\mathbf{u} = \mathbf{m}_i(\mathbf{x})$. Consequently, the sum of squared differences in intensity from all the images, $\sigma'(\mathbf{x})$, is:

$$\mu'(\mathbf{x}) = \frac{1}{n_i}\sum_{i=1}^{n_i} g_i(\mathbf{m}_i(\mathbf{x}))$$

$$\sigma'(\mathbf{x}) = \frac{1}{n_i}\sum_{i=1}^{n_i} \left(g_i(\mathbf{m}_i(\mathbf{x})) - \mu'(\mathbf{x})\right)^2$$

Figure 3(a) illustrates the projection of a sample-point of a facet onto several images.

The above definition of $\sigma'(\mathbf{x})$ does not take into account occlusions of the surface. To do so, we use a "Facet-ID" image shown in Figure 4. It is generated by encoding the index i of each facet f_i as a unique color, and projecting the surface into the image plane using a standard hidden-surface algorithm. Thus, when a sample-point from facet f_k is projected into an image, the index k is compared to the index stored in the Facet-ID image at that point. If they are the same, then the sample-point is visible in that image, otherwise, it is not. Let $v_i(\mathbf{x}) = 1$ when point $\mathbf{x}$ is determined to be visible in image g_i by the method above, and $v_i(\mathbf{x}) = 0$ otherwise. Then, the correct form for the sum of squared differences in intensity at a point $\mathbf{x}$ is:

$$\mu(\mathbf{x}) = \frac{\sum_{i=1}^{n_i} v_i(\mathbf{x}) g_i(\mathbf{m}_i(\mathbf{x}))}{\sum_{i=1}^{n_i} v_i(\mathbf{x})}$$

$$\sigma(\mathbf{x}) = \frac{\sum_{i=1}^{n_i} v_i(\mathbf{x}) \left(g_i(\mathbf{m}_i(\mathbf{x})) - \mu(\mathbf{x})\right)^2}{\sum_{i=1}^{n_i} v_i(\mathbf{x})}$$

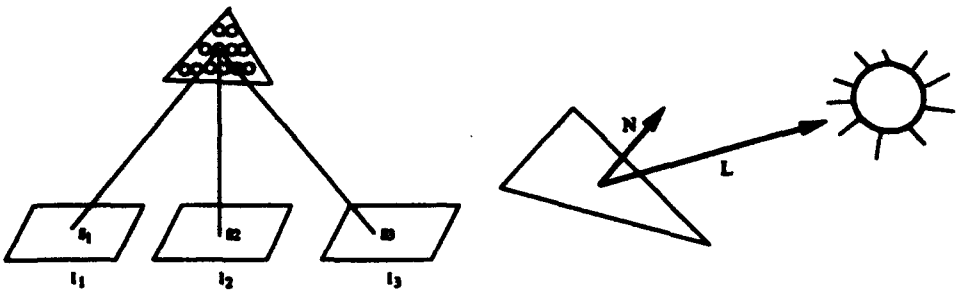
Finally, summing $\sigma(\mathbf{x})$ over all sample-points and over all facets yields the multi-image intensity correlation component:

$$\mathcal{E}_C(\mathcal{S}) = \sum_{k=1}^{n_f} c_k \sum_{l=3}^{n_s} \sigma(\mathbf{x}_{k,l}),$$

where c_k is a number between 0 and 1 that weights the contribution from each facet differently, depending on the average degree of texturing within a facet (see Section 4.3.4).

When the original surface giving rise to the images is sufficiently textured, this component should be smallest when the surface $\mathcal{S}$ closely approximates the original surface. However, when the surface has constant, or nearly constant, albedo this component would be small for many different surfaces. As an extreme example of this ambiguity, consider a planar surface with constant albedo. This produces images with constant intensity. Thus, this component will not be able to constrain the shape of the surface, since the difference in intensity will be zero for all surfaces.

An example of using only the intensity-correlation and smoothness components on the synthetic stereo pair of Figure 1 is shown in Figure 5. The top row of the figure depicts the initial surface estimate. Figures 5(a) and (b) are shaded images of the mesh. Figure 5(c) depicts the error from ground-truth elevation for the left image, where black indicates zero error, and white indicates an error corresponding to a few pixels in disparity. Figure 5(d) depicts the squared difference in intensity between the left image and the right images warped using the disparity map. Note that the worst errors occur



(a) (b)

Figure 3: (a) Facets are sampled at regular intervals as illustrated here. We use the grey levels of the projections of these sample points to compute the stereo score. (b) The albedo of each facet is estimated using the facet normal $\vec{N}$, the light source direction $\vec{L}$ and the average grey level of the projection of the facet into the images.

(a) (b) (c)

Figure 4: Illustration of the projection of a mesh, and the "Facet-ID" image used to accomodate occlusions during surface reconstruction. (a) A shaded image of a mesh. (b) A wire-frame representation of the mesh (bold white lines) and the sample-points in each facet (interior white points). (c) The "Facet-ID" image, wherein the color at a pixel is chosen to uniquely identify the visible facet at that point (shown here as a grey-level image).

along the steep ridge of the terrain, where the constant-depth assumption of correlation-based stereo is most strongly violated.

The bottom row of Figure 5 illustrates the result of the optimization procedure, described in Section 4.4, using only the intensity-correlation and smoothness components. Note that the overall error in both elevation and intensity is lower, and that the error is no longer concentrated along the ridge. As a result, the ridge is clearly sharper in the shaded views.

4.3.3 Shading

The shading component of the objective function is the sum, over all facets, of the difference between the computed albedo of the facet and the computed albedoes of all of its neighbors. The motivation for this component, and its precise form, follow.

Recall that the Lambertian reflectance model defines the intensity g at a point on a surface with a unit surface normal $\vec{N}$ as:

$$g = \alpha(a + b\vec{N} \cdot \vec{L}), \qquad (3)$$

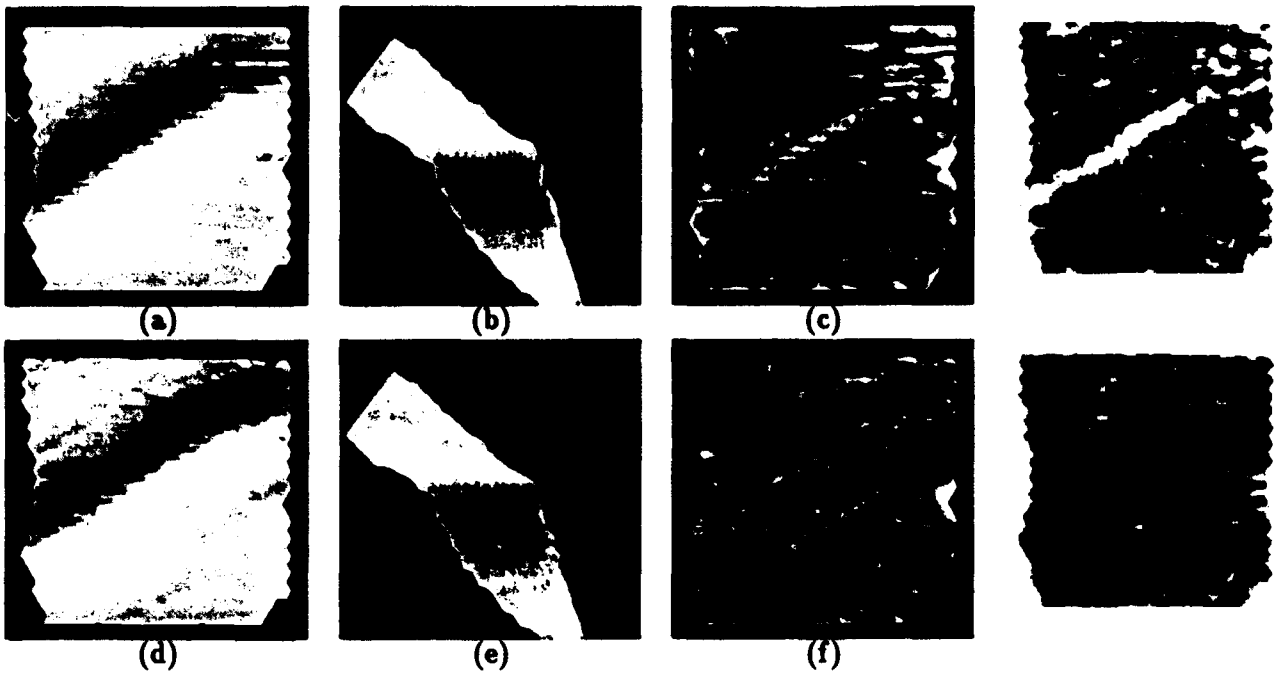


Figure 5: (a,b) Two shaded views of the mesh derived from the smoothed disparity map of Figure 1(d). (c) Deviations in altitude from the elevation data used to generate the synthetic pair. (d) Intensity error image, created by warping the right image into the left image using the disparities corresponding to the elevations of the mesh facets and computing the squared difference between these two images (e,f,g,h) Corresponding images after stereo optimization. Note that the ridge now appears much sharper in the shaded views, and that the overall error is smaller and more evenly distributed.

where α is the albedo of the surface, a is the magnitude of the ambient light, b is the magnitude of a point light source, and $\vec{L}$ is the direction of the point light source as depicted in Figure 3(b).

Note that g is independent of the viewing direction. Consequently, if we were to image a planar Lambertian facet from several points of view, its intensity would be the same for all pixels in the projection of the facet. Conversely, if we were to measure the average intensity $\bar{g}_k$ of all of the pixels within the projection of a facet f_k, we could compute its albedo, α_k, as follows:

$$\alpha_k = \frac{\bar{g}_k}{(a + b\vec{N} \cdot \vec{L})}. \qquad (4)$$

This assumes, of course, that the facet is well-modeled by a single albedo, and that the variation in intensity is due only to noise. In this paper, we assume that the ambient and direct illumination (*i.e.*, a, b, and $\vec{L}$) are given, although some of these parameters could be included in the optimization, as was done in [31].

The average intensity $\bar{g}_k$ of a facet is computed by scanning over all the Facet-ID images for index k, and taking the average of the intensities at matching points in the corresponding images. This method provides an inexpensive way of computing the average intensity while taking occlusions into account.

Now, if the original surface had exactly constant albedo, and if our mesh surface were a good approximation to the original surface, then the computed albedoes should be approximately the same across all facets. Thus, some measure of the variation in computed albedoes would be a good measure of the correctness of the mesh surface. If the albedo varies slowly across the surface, we propose that an appropriate measure of this variation is the difference between the computed albedo at the facet and the computed albedoes of all of its neighbors:

$$\mathcal{E}_S(\mathcal{S}) = \sum_{k=1}^{n_f}(1-c_k)\sum_{j\in N_f(k)}(1-c_j)(\alpha_k-\alpha_j)^2,$$

where $N_f(k)$ is the set of indices of the facets that are neighbors of facet f_k, and c_k and c_j are numbers between 0 and 1 that depend on the degree of texturing within facets f_k and f_j.

An example of using only the shading and smoothness components is illustrated in Figure 6. Figure 6(a) shows a shaded view of the original surface, a hemisphere with constant albedo. Figures 6(b) and (c) show shaded views of the initial surface estimate, which was derived by adding white noise to the vertex coordinates of the original surface. Figures 6(d) and (e) are the shaded views of the result after optimization, and Figure 6(f) is the albedo map for the surface, i.e. the intensity in the image represents the albedo of the surface. Note that the albedo and shape are well recovered except near the edge of the hemisphere where the image intensity varies rapidly across the image. This is because the approximation we use in the derivatives of this component is that the mean intensity within a facet does not vary significantly in the neighborhood of a facet, which is violated for facets that straddle the boundary. This does not hurt us when combining shading with the stereo component since, as explain in the following subsection, we turn off the shading component in such areas.

4.3.4 Combining the Components

Recall that the objective function $\mathcal{E}(\mathcal{S})$ is a linear combination of three components:

$$\mathcal{E}(\mathcal{S}) = \lambda_D\mathcal{E}_D(\mathcal{S}) + \lambda_C\mathcal{E}_C(\mathcal{S}) + \lambda_S\mathcal{E}_S(\mathcal{S}),$$

where the last two components are themselves linear combinations of subcomponents computed on a per-facet basis:

$$\mathcal{E}_C(\mathcal{S}) = \sum_{k=1}^{n_f} c_k \sum_{l=3}^{n_s}\sigma(\mathbf{x}_{k,l}) \qquad (5)$$

$$\mathcal{E}_S(\mathcal{S}) = \sum_{k=1}^{n_f}(1-c_k)\sum_{j\in N_f(k)}(1-c_j)(\alpha_k-\alpha_j)^2.$$

Thus, one needs to specify both the λs, defining the relative weights of the components, and the c_ks, defining the relative weights of the facets in each of these components.

The λ weights are defined as follows:

$$\lambda_D = \frac{\lambda'_D}{\| \vec{\nabla}\mathcal{E}_D(\mathcal{S}^0) \|}$$

$$\lambda_C = \frac{\lambda'_C}{\| \vec{\nabla}\mathcal{E}_C(\mathcal{S}^0) \|} \qquad (6)$$

$$\lambda_S = \frac{\lambda'_S}{\| \vec{\nabla}\mathcal{E}_S(\mathcal{S}^0) \|},$$

where $\mathcal{S}^0$ is the initial estimate of the surface, and the λ's are user defined weights. Normalizing each component by the magnitude of its initial gradient allows the components to have roughly the same influence when the λ's are equal. Thus, the user can more easily specify the relative contributions of each component in an image-independent fashion. This normalization scheme was used with great success in [16], and is analogous to standard constrained optimization techniques in which the various constraints are scaled so that their eigenvalues have comparable magnitudes [34].

As mentioned earlier, the c_k weights are a function of the degree of texturing in the intensities projected within a facet f_k. A simple measure of the degree of texturing within a facet is the variance in intensity of all the pixels projecting onto the facet, denoted $\sigma_k(\mathcal{S})$ (using the Facet-ID image to accommodate occlusions). We have found that using the logarithm of $\sigma_k(\mathcal{S})$ yields the most stable results:

$$c_k = a\log(1+\sigma_k(\mathcal{S})) + b, \qquad (7)$$

where a and b are normalizing factors chosen so that the smallest c_k is zero, and the largest is one.

4.4 The Optimization Procedure

The purpose of the optimization procedure is to iteratively modify the surface $\mathcal{S}$ so as to minimize $\mathcal{E}(\mathcal{S})$, given some initial estimate $\mathcal{S}^0$, and some value for the weights λ'_S, λ'_C, and λ'_D

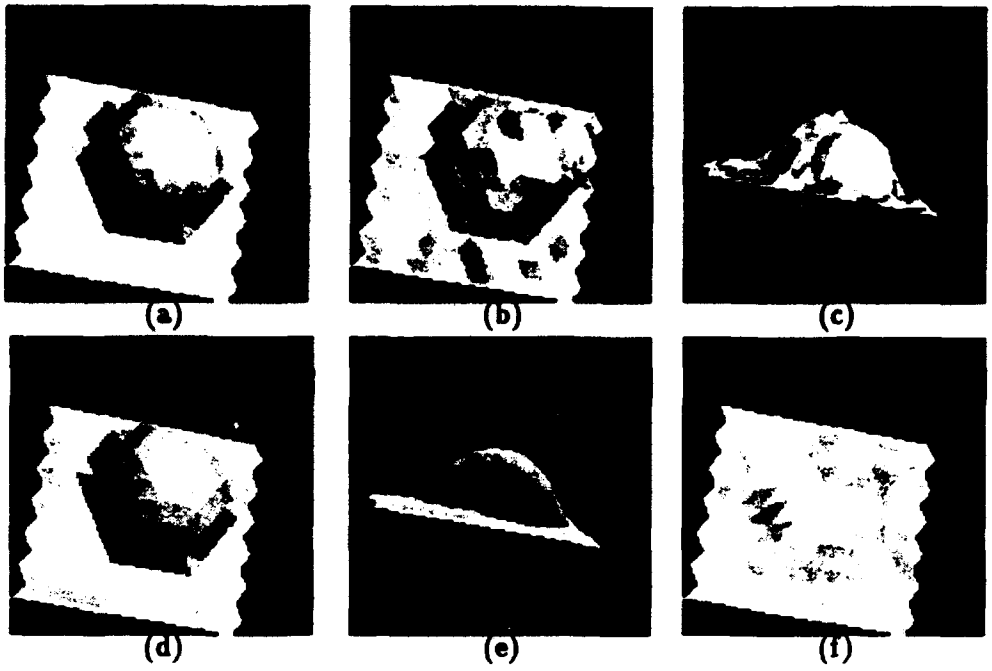


Figure 6: (a) Shaded image of a hemisphere of contant albedo. (b,c) Shaded views of randomized hemisphere used as a starting point. (d,e) Shaded views of the same hemisphere after optimization using only the shading component of the objective function. (f) The recovered albedo map.

(where $\lambda'_S + \lambda'_C + \lambda'_D = 1$) defined in Equation 7. Ideally, one would like to use as small a value of the deformation weight λ'_D as possible so as to minimize the bias introduced by this term. However, in practice, λ'_D serves a dual purpose. First, since the surface deformation term is a quadratic function of the vertex coordinates, it "convexifies" the energy landscape and improves the convergence properties of the optimization procedure. Second, as will be discussed in the results section, in the absence of a smoothing term, the objective function may overfit the data and wrinkle the surface excessively. Furthermore, the c_k weights of Equations 6 and 7 are computed for the initial position of the mesh and are only meaningful when it is relatively close to the actual surface.

Consequently, we use an optimization method that is inspired by the heuristic technique known as a continuation method [43, 28, 29, 30]. We first "turn off" the shading term by setting λ'_S (equation 7) to 0 and set λ'_D to a value that is large enough to sufficiently convexify the energy landscape but small enough to allow curvature in the surface. In this paper we take the initial value of λ'_D to be 0.5. Given the initial estimate $\mathcal{S}^0$, a local minimum of this approximate objective function is found using a standard optimization procedure. Then, λ'_D is decreased slightly, and the optimization procedure is applied again, starting at the local minimum found for the previous approximation. This cycle is repeated until λ'_D is decreased to the desired value. Finally we "turn on" the shading term, compute the c_k weights and reoptimize. In all examples shown in the result section we use $\lambda'_C = \lambda'_S = .4$ and $\lambda'_D = .2$.

The stereo component effectively uses only first order information about the surface (i.e., the position of the vertices), whereas shading uses second order information about the surface (i.e., its surface normals). Thus, by optimizing the stereo component first, we effectively compute the zero order properties of the surface and set up boundary conditions that the shading component can then use to compute the first order properties of the surface in textureless regions. In section 5, we will show that this

leads to a significant improvement over using the stereo component alone.

When dealing with surfaces for which motion in one direction leads to more dramatic changes that motions in others, as is typically the case with the z direction in Digital Elevation Models (DEMs), we have found that the following heuristic to be useful. We first fix the x and y coordinates of vertices and adjust z alone. Once the surface has been optimized, we the allow all of the coordinates to vary simultaneously.

The optimization procedure we use at every stage is a standard conjugate-gradient descent procedure called FRPRMN (from [38]) in conjunction with the a simple line search algorithm. The conjugate-gradient procedure requires three inputs: 1) a function that returns the value of the objective function for any $\mathcal{S}$; 2) a function that returns the gradient of $\mathcal{E}(\mathcal{S})$, *i.e.*, a vector whose elements are the partial derivatives of $\mathcal{E}(\mathcal{S})$ with respect to the vertex coordinates, evaluated at $\mathcal{S}$; and 3) an initial estimate $\mathcal{S}^0$.

The gradient of $\mathcal{E}(\mathcal{S})$ is conceptually straightforward, but is fairly complicated to derive manually. We have used the Maple [1] mathematical package to derive some of the terms. We summarize the calculation of the derivatives below in general terms.

The derivatives of the stereo term are linear combinations of image intensity derivatives and of derivatives of the 3-D projections of points onto the images. Since we use bilinear-interpolation of image values, the first derivatives of image intensity are linear combinations of the image intensities in the immediate neighborhood of the projection. Since sample-points are linear combinations in projective space of the mesh vertices, their projections are ratios of linear combinations of the projections of the vertices, which themselves depend linearly on the vertex coordinates. Consequently, the derivatives of these projections are ratios of linear combinations of the vertex coordinates and squares of linear combinations of the vertex coordinates.

Similarly, the derivatives of the shading term depend of the derivatives of the surface normal, which can be easily derived analytically, and from the derivative of the mean grey-level in the facets. In this work, the shading term is used mainly in the fairly uniform areas where the latter derivative is assumed to be small and therefore neglected.

[1] Trademark, Waterloo Maple Software

5 Behavior of the Objective Function and Results

In previous sections, we have shown results of the optimization procedure using only one or the other of the image components of the objective function. In this section, we first illustrate the behavior of the complete objective function using synthetic data. We then show that the same behavior can be observed with real data, allowing us to generate accurate 3-D reconstructions of real surfaces from multiple images.

5.1 Synthetic Data

To demonstrate the properties of the objective function of Equation 1 and the influence of the coefficients defined in Equation 4, we use as input the five synthetic images of a shaded hemisphere with variable albedo shown at the bottom of Figure 7, both with and without the addition of white noise. Each column of the figure illustrates the steps used in the creation of the image at the bottom of the column. We begin with a mesh and an albedo map, shown in the top row. Then, for each view, two images are produced. The first image (second row of the figure) is the albedo map texture-mapped onto the mesh from the final image's point of view. The second image (third row of the figure) is a shaded view of the mesh, using a constant albedo equal to one. The final image is the point-by-point product of these two images because, by Equation 3, the imaged intensity of a Lambertian surface is the product of the albedo (first image) and the inner product of the light source and the surface normal (second image).

Figure 8 depicts graphically the result of our experiments. In each experiment we randomized the mesh by adding random numbers to the coordinates of the mesh vertices, and added different amounts of noise to the input images.

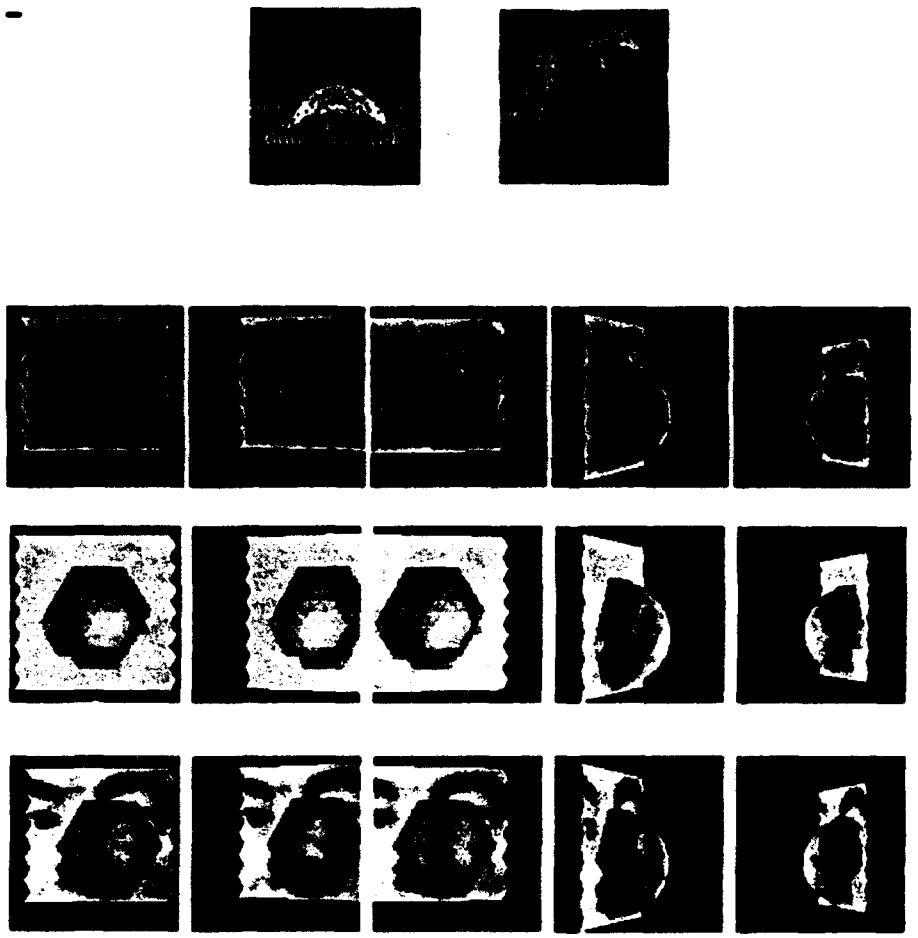


Figure 7: The making of synthetic images of a shaded hemisphere with variable albedo that conforms to our Lambertian model.

We then used our optimization procedure to estimate the true hemispherical shape and true albedo map. More precisely, starting from our randomized initial estimate, we first use stereo alone and progressively decrease the value of the λ'_D parameter of Equation 7 from 0.5 to 0. We then turn on the shading term by setting both λ'_D and λ'_S to 0.4, compute the c_ks of Equation 7 and optimize the full objective function. To show the stability of the process, we then recompute the c_ks for the optimized mesh and perform a second optimization using the updated values.

The first column of Figure 8 is for experiments using only the first, second, and third images from Figure 7, where there is little self occlusion. The second column is for experiments using the first, fourth, and fifth images, where there is a significant amount of self occlusion. Finally, the third column is for experiments using all five images. In this particular set of experiments, we fixed the boundaries of the mesh and allowed only the z coordinates of the vertices to vary. However, the same overall behaviors can be observed without the boundary conditions.

The first row from the top of Figure 8 is a graph of the average squared error in elevation (the abscissa) versus decreasing λ'_D (the ordinate). To the left of the dotted vertical line, only the intensity correlation component is used. To the right, both the intensity correlation and shading components are used. The different curves are for different amounts of noise in the input images. The bottom curve is when there is no noise (other than quantization error),

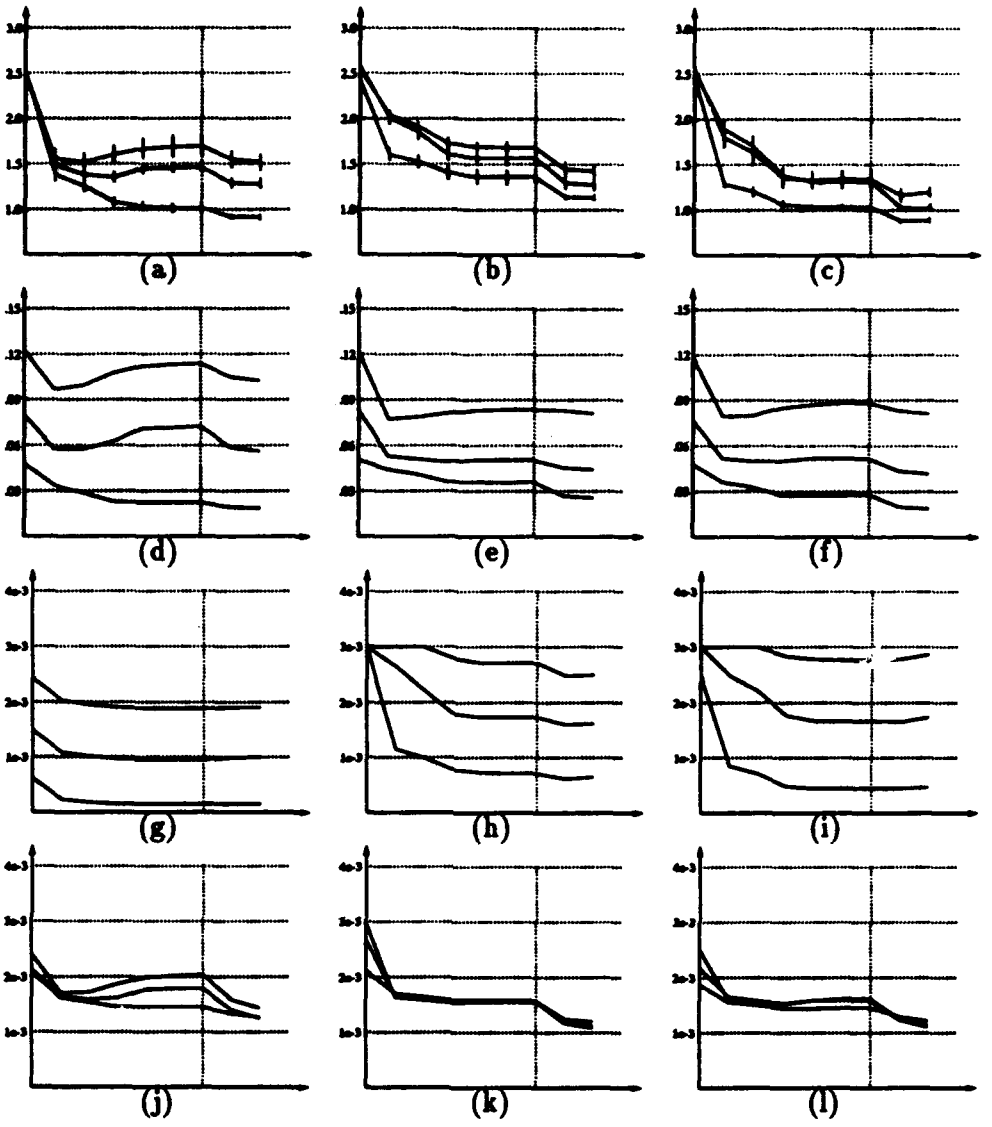


Figure 8: Graphs of the errors and objective function components while fitting a surface model to the synthetic shaded hemisphere images of Figure 7 (These graphs are explained in detail in the text.). (a,b,c) Average error in recovered elevation. (d,e,f) Average error in recovered albedo. (g,h,i) Stereo component of the energy. (j,k,l) Shading component of the energy.

the middle curve is for a noise variance of 4% of the image dynamic range, and the top curve is for a noise variance of 8%. The short vertical lines along the curves indicated the standard deviation of the average error over the 20 experiments performed to derive each curve.

The second row of Figure 8 is a graph of the average error in computed albedo. The third row is the average value of the intensity correlation component, $\mathcal{E}_C(\mathcal{S})$, and the fourth row is the average value of the shading component, $\mathcal{E}_S(\mathcal{S})$.

Note that, as λ'_D decreases and stereo alone is used (*i.e.*, as the ordinate is traversed rightwards to the dotted vertical line), the average elevation error decreases when there is no noise in the input image (bottom curve), as does the average albedo error and the two components of the objective function. However, when the images are noisy, the elevation error (first row) stops decreasing and may even begin to increase as we start fitting to the grey-level

noise, even though the value of the intensity correlation component (third row) continues to decrease (as it must). Furthermore, both the albedo error (second row) and the shading component (fourth row) also begin to increase when the elevation error does. This is natural since for smaller values of λ'_D the surface becomes rougher and its normals less well-behaved. As a result, the estimated albedoes of Equation 4 become less reliable and noisier.

In other words, an increase in the shading component provides us with a warning that we are starting to overfit the data. This is a valuable behavior in itself. Furthermore, by turning on the shading component of our objective function (those parts of the graphs that are to the right of the vertical dotted line), we can bring down both the error in albedo and the value of albedo component with at worst of modest increase in the value of the stereo component, resulting in an overall reduction of the elevation error. Even when there is nothing but quantization noise in the image, the addition of the shading component can make a small, but still noticeable difference. The reasons for this are twofold:

1. The shading component averages over whole facets and is therefore less sensitive to uncorrelated noise.

2. The shading component uses absolute intensity values whereas the stereo component uses intensity differences. Thus, in the presence of noise in textureless areas, the signal-to-noise ratio for the absolute values (used by the shading component) is larger than for the differences (used by the stereo component), thereby making the shading term more robust.

However, in our experience, the shading term can only be used reliably when the surface is relatively close to the correct answer. This is not surprising since the stereo deals directly with elevations whereas shading deals with derivatives of elevation. Consequently we have chosen the optimization "schedule" described above where we first optimize using stereo alone and turn on shading only later.

There is another important point to note about these results. The elevation errors in the second row, i.e those generated using images 1, 4, and 5 with a lot of self occlusion are very close to those of the first row, i.e. those generated using images 1, 2, and 3 with little self occlusion, while those in the final row (using all five images) are significantly better. Furthermore, in this particular case, the results for images 1,4 and 5 are even slightly better than those for images 1,2 and 3 in the presence of noise because the former correspond to larger baselines. In other words, having the same number of images, but with significant self-occlusions, does not hurt our procedure. However, adding new images that contain significant self-occlusions actually improves the results.

We now turn to real images and show that the same properties can also be observed there.

5.2 Real Images

In Figure 9 we show the result of running the stereo component of our objective function on a real stereo pair corresponding to the same site as the synthetic images of Figure 1. Note that the radiometry of the left and right images are actually slightly different. We correct for this by first band-passing each image by taking the difference between the image and its gaussian convolution. This is approximately equivalent to replacing the simple correlation that our objective function uses by a normalized correlation, but is computationally more efficient. We then applied the optimization using exactly the same schedule and parameters as in the synthetic case, with the exception that λ_S is not reduced quite as much for the real images as for the synthetic ones in the first step of the procedure. Note that the recovered ridge is even sharper than in the synthetic case. This is because the Digital Elevation Model used to produce the synthetic right image was actually a slightly smoothed version of the terrain, in which one side of the ridge is an almost vertical cliff. Thus, even though we do not currently have ground truth for the real case, the sharpness of the recovered cliff, which matches what is seen using a stereoscope, leads us to believe that the algorithm has performed well.

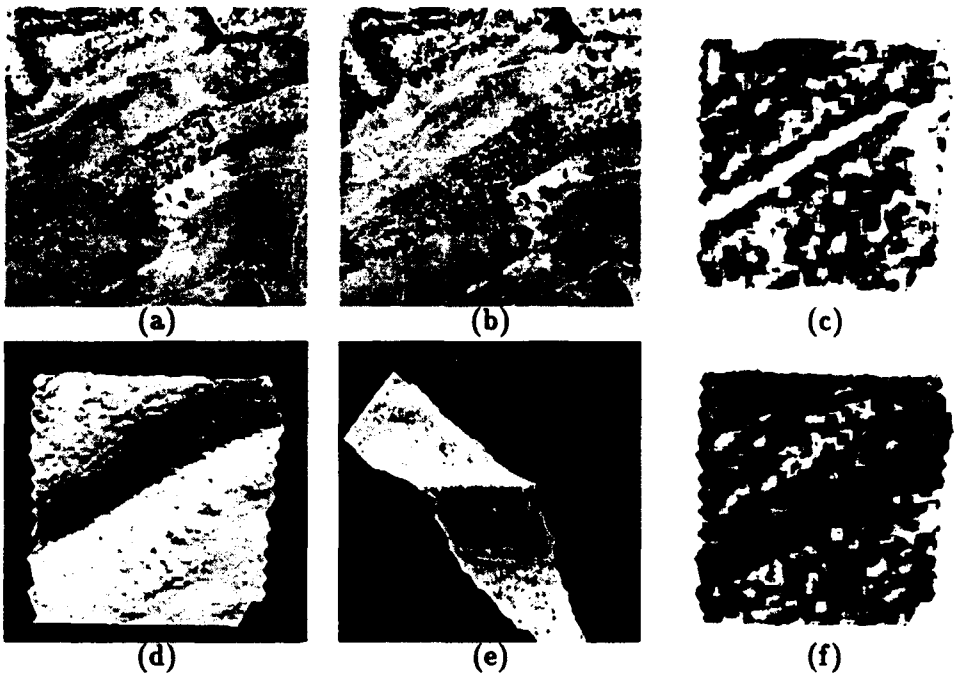


Figure 9: (a,b) A stereo pair of real images of the Martin-Marietta ALV test-site used in Figure 1. (c) Intensity error image computed using the method described in Figure 1(c) (d,e) Shaded views of the mesh after optimization. (f) Intensity error image after optimization. Note that the ridge is now very sharp. This corresponds accurately to the almost vertical cliff that can be seen when viewing the stereo pair with a stereoscope.

In Figure 10 we show three triplets of images of faces. They have been produced using the INRIA three camera system [13] that provides us with the 3 by 4 projection matrices we need to perform our computations. In this case it is essential to have more than two images to be able to reconstruct both sides of the face because of self-occlusions. For each triplet, we have computed disparity maps corresponding to images 1 and 2 and to images 1 and 3 and combined them to produce the depth maps shown in the rightmost column of the figure using the algorithms described in [19, 15].

The depth maps have then been smoothed and triangulated to produce the initial surfaces shown in the upper left corner of Figures 11, 12, and 13. In the first row of these three figures, we show the result of the optimization using stereo alone as we progressively decrease the smocthness constraint and allow all three vertex coordinates to be adjusted. Note that for the first two triplets (Figures 11 and 12), we recover more and more detail until the surface eventually starts to wrinkle, without apparent improvement in accuracy. The third triplet poses an even more difficult problem: there are strong specularities on both the forehead and the nose that strongly violate our Lambertian model. Because there are very few other points that can be matched on the nose, the algorithm latches on to these specularities and yields a poor result.

In the bottom row of Figures 11, 12, and 13, we show our final results obtained by turning on the shading term and reoptimizing the meshes. For these images we did not know apriori the light source-direction, we therefore estimated it by choosing the direction that minimizes the shading component of the objective function given the surface optimized using only the stereo component. In all three images, the main features of the faces, nose, mouth and eyes have been correctly recovered. The improvement is particularly striking in the case of the face in Figure 13. The shading component was able to achieve this result because it uses

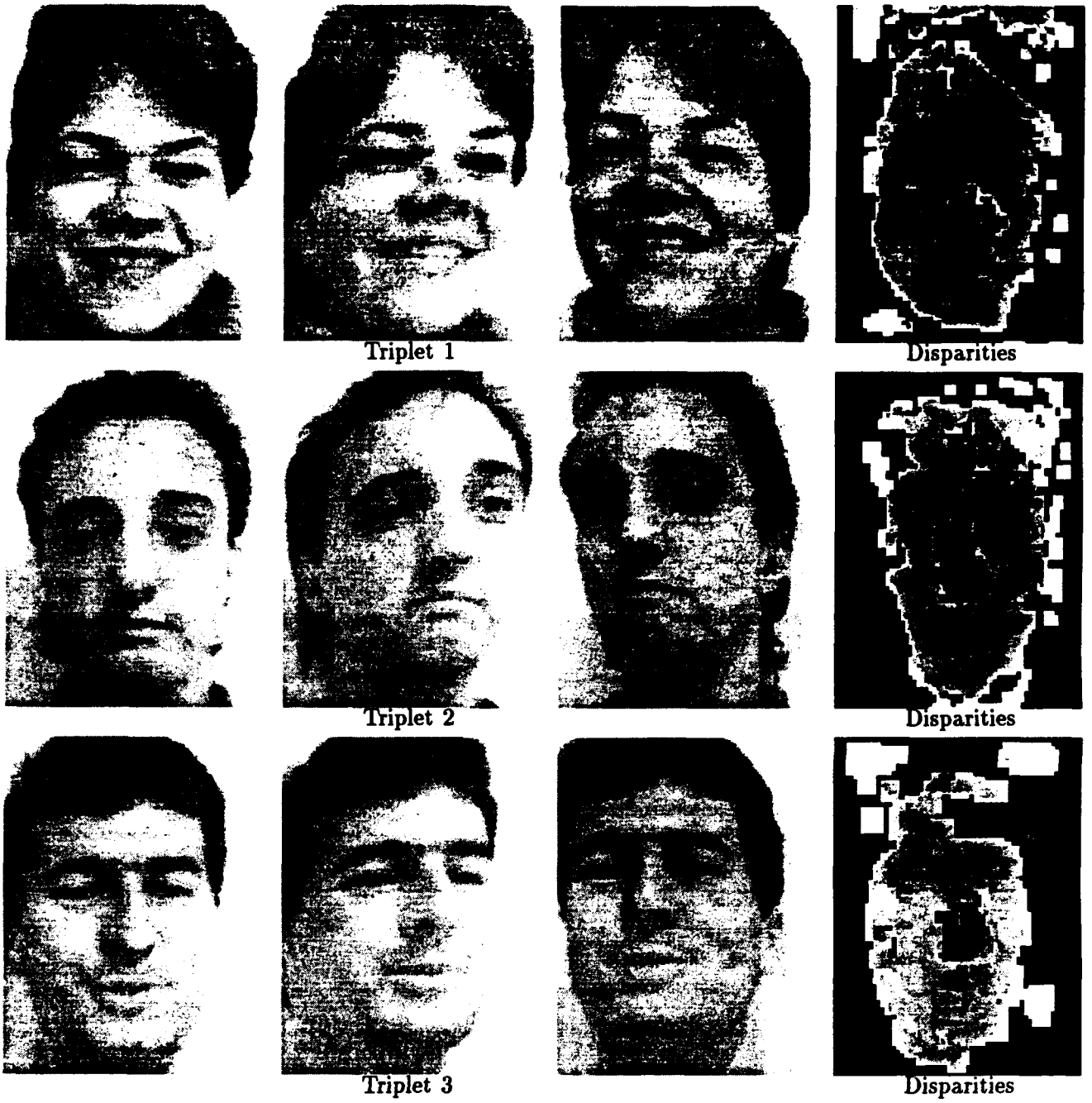


Figure 10: Triplets of face images and corresponding disparity maps (courtesy of INRIA).

the monocular information around the specularities. The stereo component cannot take advantage of the information around the specularities because very few points are visible in at least two images simultaneously, and because there is little texture. Of course, the effect of the specularities has not completely disappeared (there is indeed still a small artifact on the nose) but has been outweighed by the surrounding information. A more principled approach to solving this problem would be to explicitly include a specularity term in our shading model.

The graphs of Figure 14 depict the behavior of the stereo and shading components of the objective function for the three triplets. The four values of the scores to the left of the thick

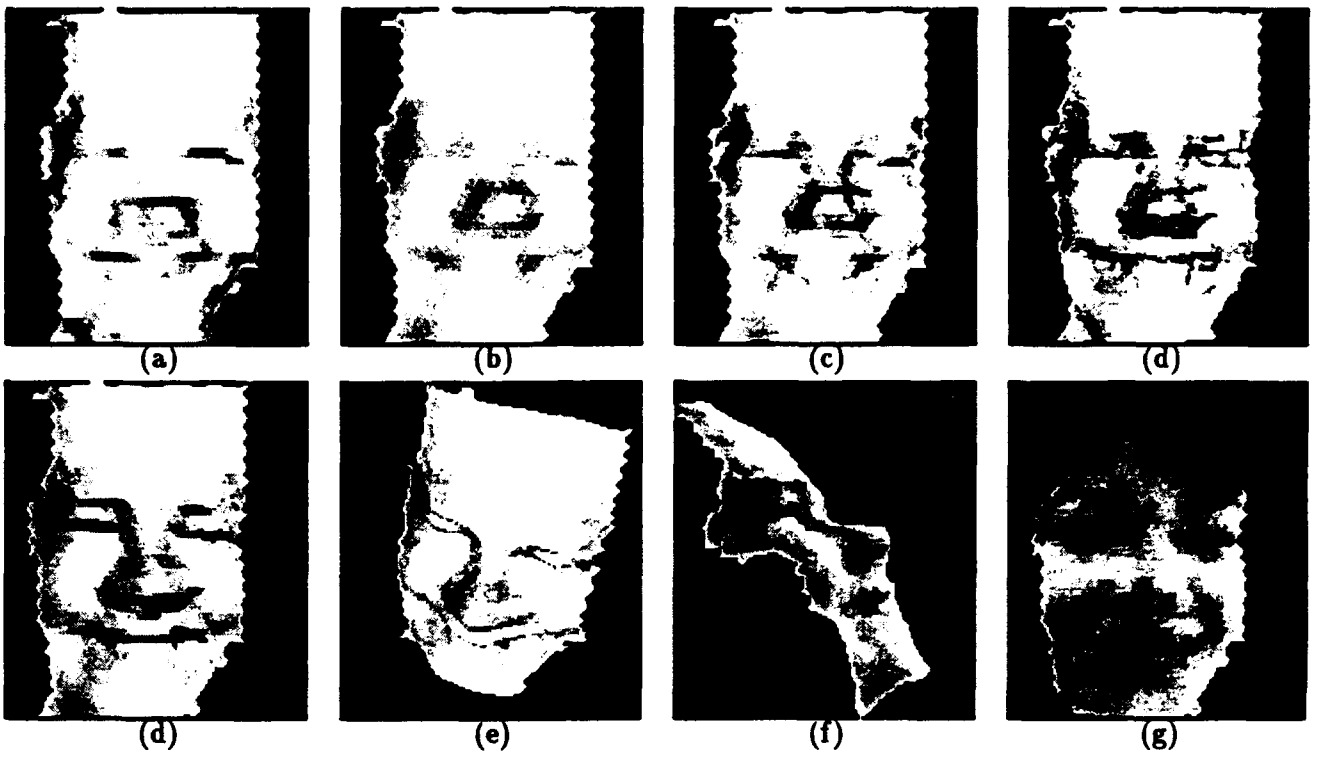


Figure 11: Results for the first triplet of Figure 10. (a) Shaded view of the mesh generated by smoothing and triangulating the computed disparity map. We use it as the starting condition for our optimization procedure. (b,c,d) The mesh after optimization using only the stereo term, with progressively less smoothing. (e,f,g) Several views of the mesh after optimization using both stereo and shading. (h) The recovered albedo map.

dotted line, St_0 to St_3, correspond to the results shown in the top row of Figures 11, 12, and 13. The fifth value, $St + Sh$, corresponds to the final results when shading is turned on. These values have been scaled so that St_0 is equal to one for all triplets. As in the synthetic case, when using stereo alone, the stereo component always improves, but as the recovered surface becomes rougher the shading term degrades dramatically. However, when we turn on the shading component, the overall results improve significantly, even though the stereo component degrades slightly.

6 Summary and Conclusion

In this paper we have presented a surface reconstruction method that uses an object-centered representation (a triangulated mesh) to recover geometry and reflectance properties from multiple images. It allows us to handle self-occlusions while merging information from several viewpoints, thereby allowing us to eliminate blindspots and making the reconstruction more robust where more than one view is available. The reconstruction process relies on both monocular shading cues and stereoscopic cues. We use these cues to drive an optimization procedure that takes advantage of their respective strengths while eliminating some of their weaknesses.

Specifically, stereo information is very robust in textured regions but potentially unreliable elsewhere. We therefore use it mainly in such areas by weighting the stereo component most strongly for facets of the triangulation that project into textured image areas. The component compares the grey-levels of the points in all of the images for which the projection of a given point on the surface is visible, as determined using a hidden-surface algorithm. This comparison is done for a uniform sampling of

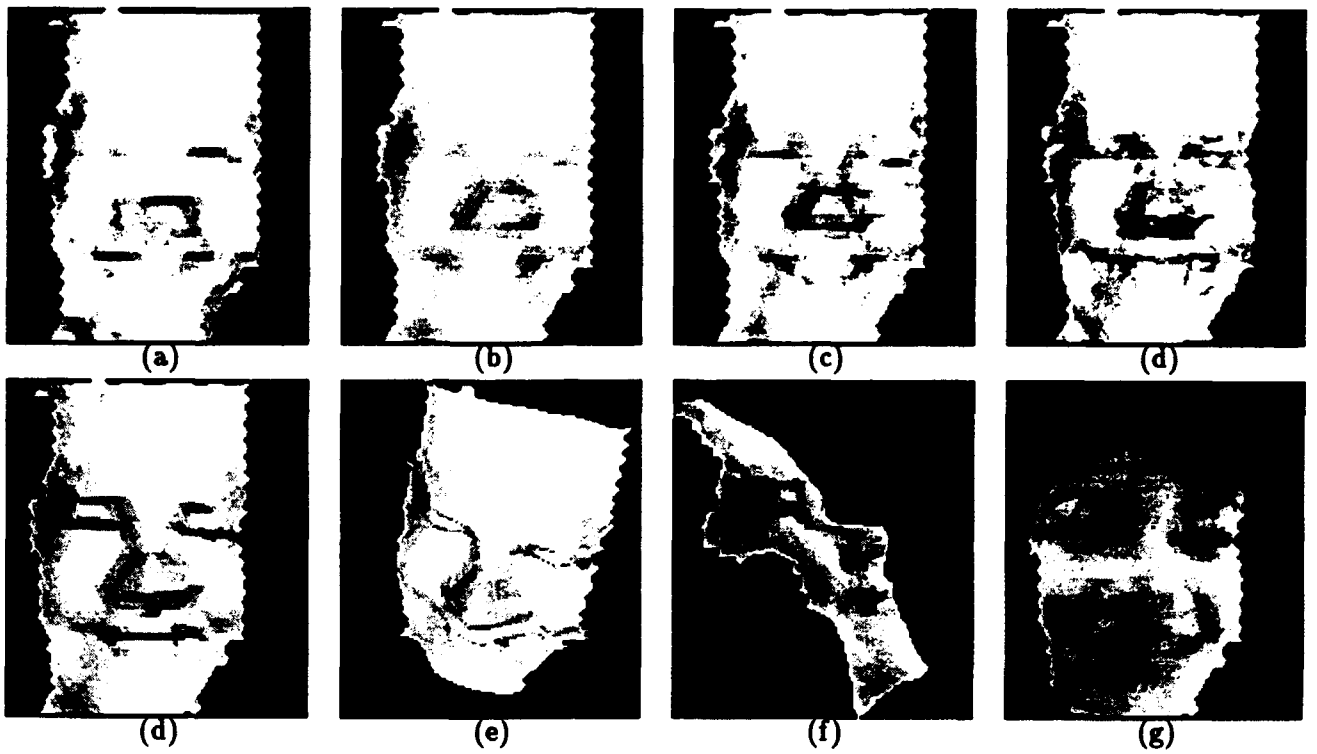


Figure 11: Results for the first triplet of Figure 10. (a) Shaded view of the mesh generated by smoothing and triangulating the computed disparity map. We use it as the starting condition for our optimization procedure. (b,c,d) The mesh after optimization using only the stereo term, with progressively less smoothing. (e,f,g) Several views of the mesh after optimization using both stereo and shading. (h) The recovered albedo map.

dotted line, St_0 to St_3, correspond to the results shown in the top row of Figures 11, 12, and 13. The fifth value, $St + Sh$, corresponds to the final results when shading is turned on. These values have been scaled so that St_0 is equal to one for all triplets. As in the synthetic case, when using stereo alone, the stereo component always improves, but as the recovered surface becomes rougher the shading term degrades dramatically. However, when we turn on the shading component, the overall results improve significantly, even though the stereo component degrades slightly.

6 Summary and Conclusion

In this paper we have presented a surface reconstruction method that uses an object-centered representation (a triangulated mesh) to recover geometry and reflectance properties from multiple images. It allows us to handle self-occlusions while merging information from several viewpoints, thereby allowing us to eliminate blindspots and making the reconstruction more robust where more than one view is available. The reconstruction process relies on both monocular shading cues and stereoscopic cues. We use these cues to drive an optimization procedure that takes advantage of their respective strengths while eliminating some of their weaknesses.

Specifically, stereo information is very robust in textured regions but potentially unreliable elsewhere. We therefore use it mainly in such areas by weighting the stereo component most strongly for facets of the triangulation that project into textured image areas. The component compares the grey-levels of the points in all of the images for which the projection of a given point on the surface is visible, as determined using a hidden-surface algorithm. This comparison is done for a uniform sampling of

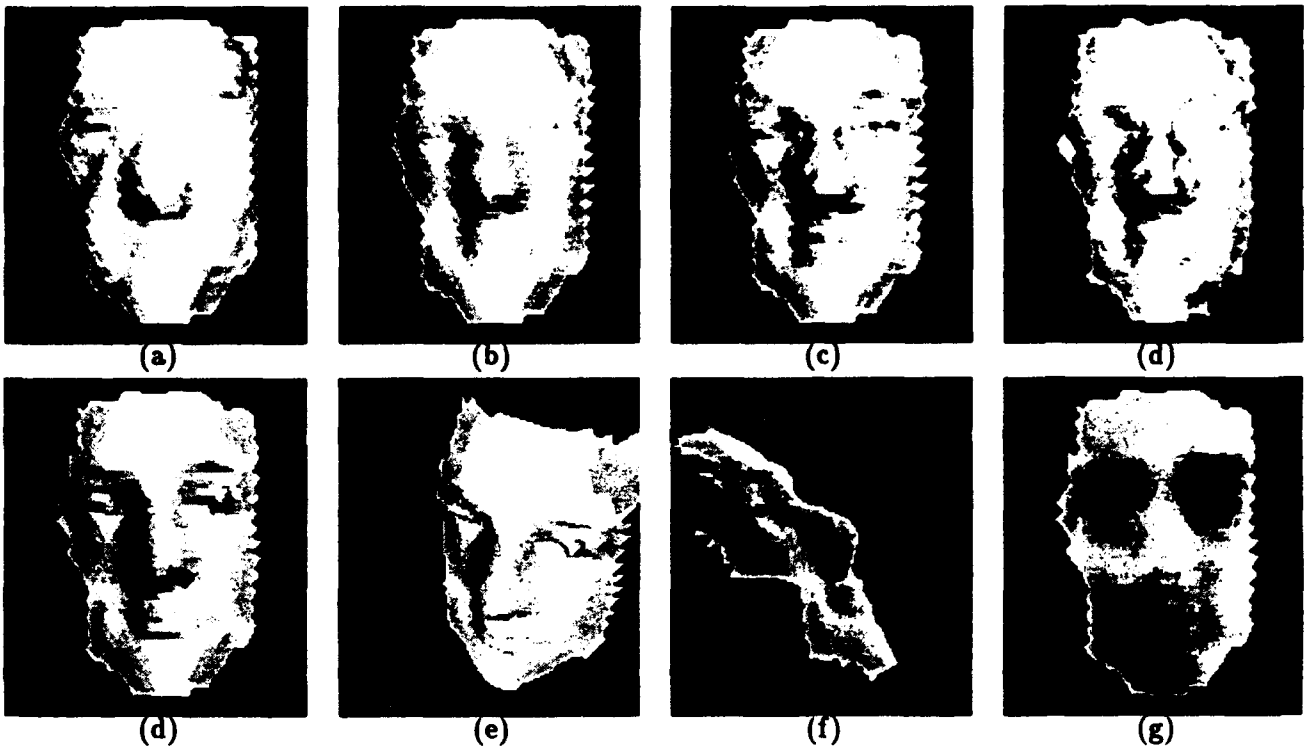


Figure 12: Results for the second triplet of Figure 10 presented in the same fashion as in Figure 11.

the surface. This method allows us to deal with arbitrarily slanted regions and to discount occluded areas of the surface.

On the other hand, shading information is mostly helpful in textureless areas. Thus, we weight the shading component most strongly for facets that project into such areas. The component uses a new method for utilizing shading information that does not need the traditional assumption of constant albedo. Instead, it attempts to minimize the variation in albedo across the surface, and can therefore deal with both constant albedo surfaces and surfaces whose albedo varies slowly. However, it does require the boundary conditions that are provided by the stereo information.

We have developed a weighting scheme that allows our system to use each source of information where it is most appropriate. As a result, for the large class of surfaces that roughly satisfy the Lambertian model, it performs significantly better than if it were using either source of information alone.

Our surface model can be naturally augmented to include specularities, shadows and self-shadows. It can also support more complex topologies, multiple resolutions and the shrinking or growing of the surface of interest, though in this paper we concentrated on a better understanding of the behavior of the objective function. These extensions will be the subject of future work.

Acknowledgments

We wish to thank Hervé Matthieu and Olivier Monga who have provided us with the face images and corresponding calibration data that appear in this paper that have proved extremely valuable to our research effort. We would also like to apologize to the members of the INRIA ROBOTVIS project whose faces we have mercilessly deformed during the development of the algorithms discussed above.

References

[1] A. L. Abbot and N. Ahuja. Active surface reconstruction by integrating focus, vergence,

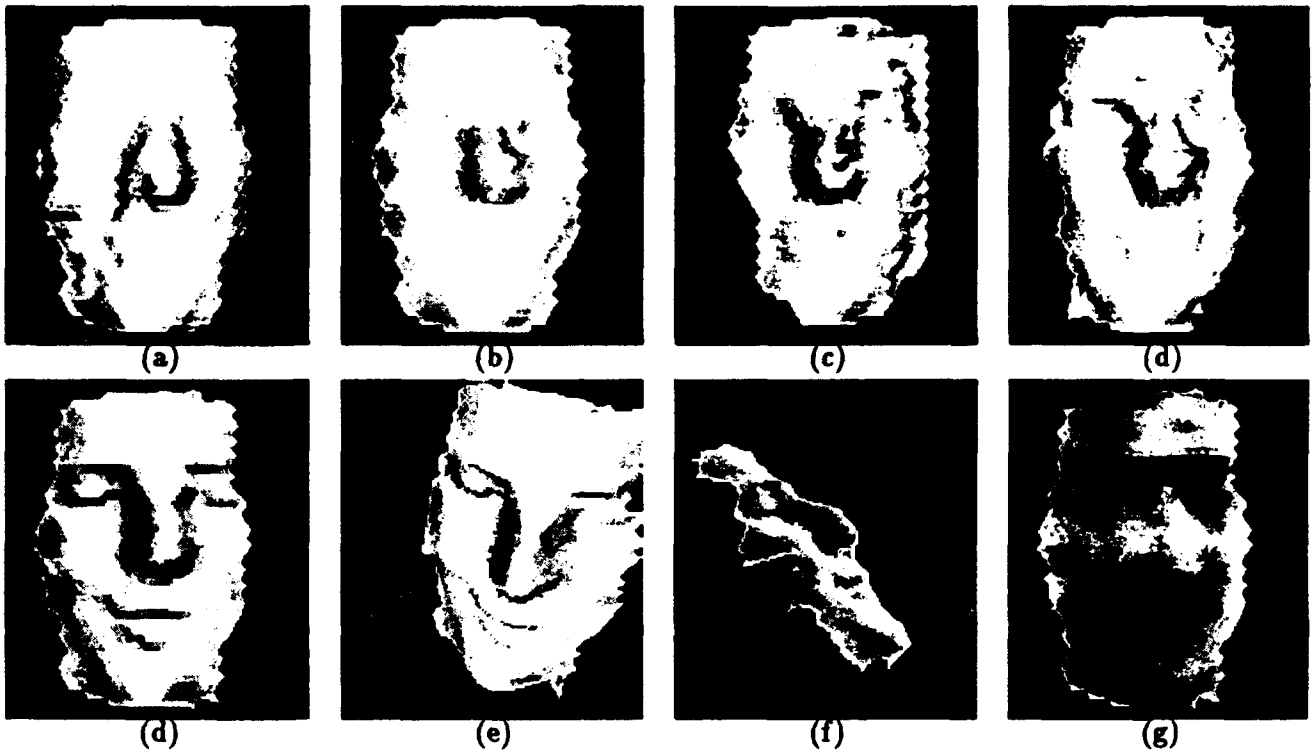


Figure 13: Results for the third triplet of Figure 10 presented in the same fashion as in Figure 11.

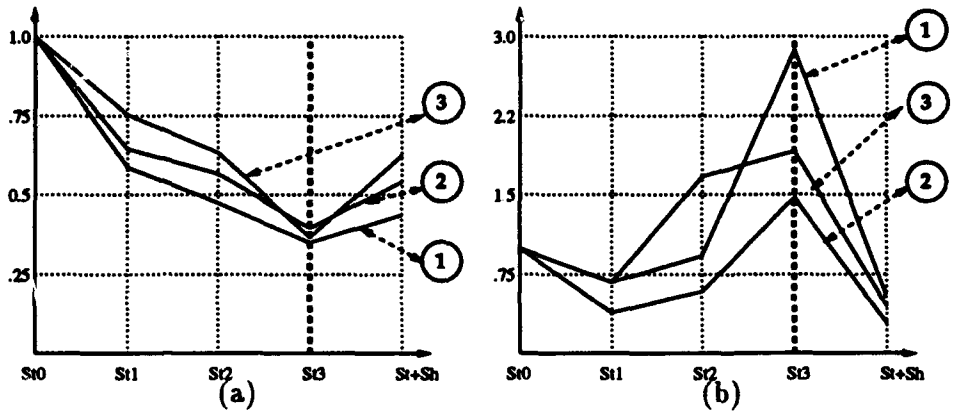


Figure 14: Values of the stereo (a) and shading (b) components of the objective function for the face images. The y axis represents the value of the components and the x axis the various stages of the optimization. From left to right, we first use only stereo and decrease the smoothness and, to the right of the thick dotted line, we turn on the shading term. Each curve is labeled with the number of responding image triplet and all values have been scaled so that the initial ones are equal to 1.0.

stereo, and camera calibration. In *ICCV*, pages 489–492, 1990.

[2] J. Y. Aloimonos. Unification and integration of visual modules: an extension of the marr paradigm. In *IUW*, pages 507–551, 1989.

[3] M. Asada, M. Kimura, Y. Taniguchi, and Y. Shirai. Dynamic integration of height maps into a 3d world representation from range image sequences. *IJCV*, 9(1):31–54, October 1992.

[4] E. P. Baltsavias. *Multiphoto Geometrically Constrained Matching*. PhD thesis, Institute for Geodesy and Photgrammetry, ETH Zurich, December 1991.

[5] S. Barnard. Stochastic stereo matching over scale. *Int'l J. Computer Vision*, 3(1):17–32, 1989.

[6] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages ?–26. Academic Press, New York, New York, 1978.

[7] A. Blake, A. Zisserman, and G. Knowles. Surface descriptions from stereo and shading. *Image Vision Comput.*, 3(4):183–191, 1985.

[8] Y. Choe and R. L. Kashyap. 3-d shape from a shaded and textural surface image. *T-PAMI*, 13:907–919, 1991.

[9] I. Cohen, L. D. Cohen, and N. Ayache. Introducing new deformable surfaces to segment 3d images. In *CVPR*, pages 738–739, 1991.

[10] J. E. Cryer, Ping-Sing Tsai, and Mubarak Shah. Combining shape from shading and stereo using human vision model. Technical Report CS-TR-92-25, U. Central Florida, 1992.

[11] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. In *CVPR*, pages 467–472, 1991.

[12] H. Diehl and C. Heipke. Surface reconstruction from data of digital line cameras by means of object based image matching. In *ISPRS*, pages 287–294, Washington D.C., 1992.

[13] O.D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In *Proceedings of CVPR86,Miami Beach,Florida*, pages 15–20, 1986.

[14] Frank P. Ferrie, Jean Lagarde, and Peter Whaite. Recovery of volumetric object descriptions from laser rangefinder images. In *European Conference on Computer Vision*, Genoa, Italy, April 1992.

[15] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 1993. In print, available as INRIA research report 1369.

[16] P. Fua and Y. G. Leclerc. Model driven edge detection. *Machine Vision and Applications*, 3:45–56, 1990.

[17] P. Fua and P. Sander. Reconstructing surfaces from unstructured 3d points. In *Proceedings of the 1992 DARPA Image Understanding Workshop*, San Diego, California, January 1992.

[18] P. Fua and P. Sander. Reconstructing surfaces from unstructured 3d points. In *Image Understanding Workshop*, San Diego, California, January 1992.

[19] P. V. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of IJCAI*, Sydney, Australia, August 1991.

[20] W. E. L. Grimson and D. P. Huttenlocher. Introduction to the special issue on interpretation of 3-d scenes. *T-PAMI*, 14(2):97–98, February 1992.

[21] K. Hartt and M. Carlotto. A method for shape-from-shading using multiple images acquired under different viewing and lighting conditions. In *CVPR*, pages 53–60, 1989.

[22] C. Heipke. Integration of digital image matching and multi image shape from shading. In *ISPRS*, pages 832–841, Washington D.C., 1992.

[23] W. Hoff and N. Ahuja. Surfaces from stereo: integrating feature matching, disparity estimation, and contour detection. *T-PAMI*, 11:121–136, 1989.

[24] B. K. P. Horn. Height and gradient from shading. *Int'l J. Computer Vision*, 5(1):37–75, 1990.

[25] Y. Hung, D. B. Cooper, and B. Cernuschi-Frias. Asymptotic bayesian surface estimation using an image sequence. *IJCV*, 6(2):105–132, June 1991.

[26] B. Kaiser, M. Schmolla, and B. P. Wrobel. Application of image pyramid for surface reconstruction with fast vision. In *ISPRS*, page 1, Washington, D.C., 1992.

[27] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[28] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, 1989.

[29] Y. G. Leclerc. *The Local Structure of Image Intensity Discontinuities.* PhD thesis, McGill University, Montréal, Québec, Canada, May 1989.

[30] Y. G. Leclerc and A. F. Bobick. The direct computation of height from shading. In *Proceedings of the 1991 Computer Society Conference on Computer Vision and Pattern Recognition*, Lahaina, Maui, Hawaii, June 1991.

[31] Y. G. Leclerc and A. F. Bobick. The direct computation of height from shading. In *Proceedings of the 1991 DARPA Image Understanding Workshop*, San Diego, California, January 1992.

[32] C. E. Liedtke, H. Busch, and R. Koch. Shape adaptation for modelling of 3d objects in natural scenes. In *CVPR*, pages 704–705, 1991.

[33] D. G. Lowe. Fitting parameterized three-dimensional models to images. *T-PAMI*, 13(441-450), 1991.

[34] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Menlo Park, California, second edition, 1984.

[35] D. Marr. *Vision.* W. H. Freeman, San Francisco, California, 1982.

[36] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, March 1990.

[37] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *T-PAMI*, 13:715–729, 1991.

[38] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes, the art of scientific computing.* Cambridge U. Press, Cambridge, MA, 1986.

[39] L. H. Quam. Hierarchical warp stereo. In *Proceedings of the 1984 DARPA Image Understanding Workshop*, pages 149–155, 1984.

[40] E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-d objects: five practical methods. *T-PAMI*, 14(8):833–839, August 1992.

[41] R. Szeliski. Shape from rotation. In *CVPR*, pages 625–630, 1991.

[42] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH'92)*, pages 185–194, July 1992.

[43] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:413–424, 1986.

[44] D. Terzopoulos. The computation of visible-surface representations. *T-PAMI*, pages 417–438, 1988.

[45] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *T-PAMI*, 13(703-714), 1991.

[46] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *CVPR*, pages 70–75, 1991.

[47] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models and 3d object reconstruction. *IJCV*, 1:211–221, 1987.

[48] C. Tomasi and T. Kanade. The factorization method for the recovery of shape and motion from image streams. In *Proceedings of the 1992 DARPA Image Understanding Workshop*, pages 459–472. DARPA, January 1992.

[49] B. C. Vemuri and R. Malladi. Deformable models: Canonical parameters for surface representation and multiple view integration. In *CVPR*, pages 724–725, 1991.

[50] Y. F. Wang and J. F. Wang. Surface reconstruction using deformable models with interior and boundary constraints. *T-PAMI*, 14(5):572–579, May 1992.

[51] P. Whaite and F. P. Ferrie. From uncertainty to visual exploration. *T-PAMI*, 13(1038-1049), 1991.

[52] A. W. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *International Journal of Computer Vision*, 1:133–144, 1987.

[53] B. P Wrobel. The evolution of digital photgrammetry from analytical phogrammetry. *Photogrammetric Record*, 13(77):765–776, April 1991.